

2 mil

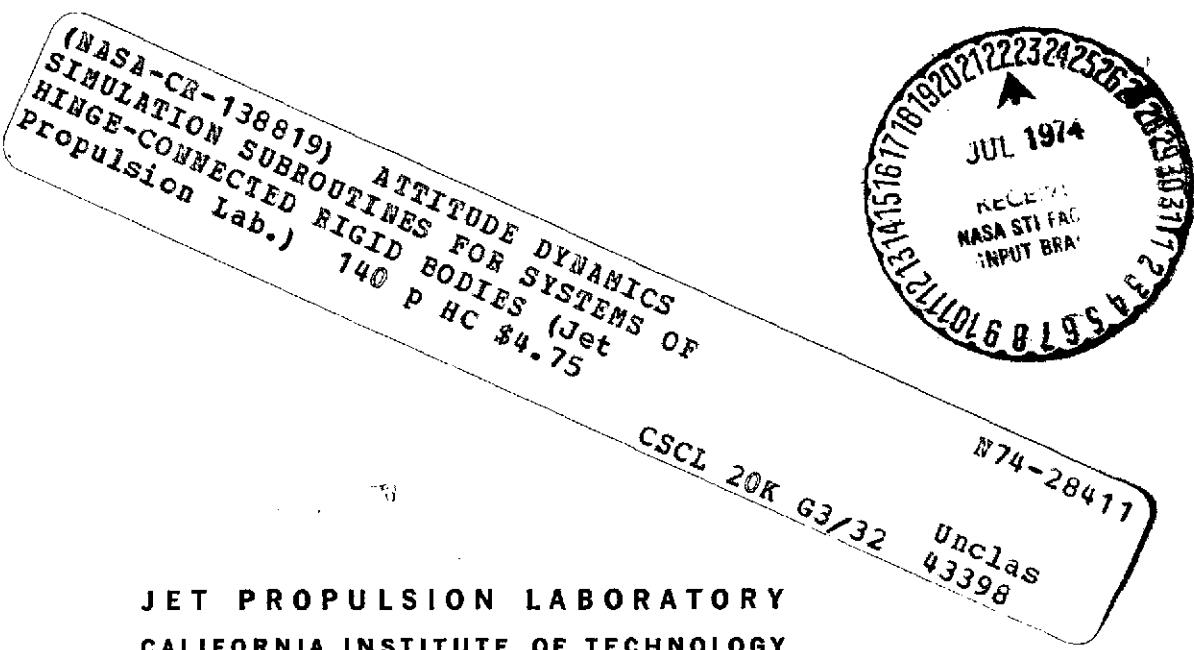
NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Report 32-1592

*Attitude Dynamics Simulation Subroutines
for Systems of Hinge-Connected
Rigid Bodies*

G. E. Fleischer

P. W. Likins



JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

May 1, 1974

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Technical Report 32-1592

*Attitude Dynamics Simulation Subroutines
for Systems of Hinge-Connected
Rigid Bodies*

G. E. Fleischer

P. W. Likins

JET PROPULSION LABORATORY
CALIFORNIA INSTITUTE OF TECHNOLOGY
PASADENA, CALIFORNIA

May 1, 1974

Prepared Under Contract No. NAS 7-100
National Aeronautics and Space Administration

11

Preface

The work described in this report was performed by the Guidance and Control Division of the Jet Propulsion Laboratory.

Errata for Technical Report 32-1565

Since the users of this report will certainly be referring to JPL TR 32-1565 (Ref. 9) for a much more detailed treatment of hinge-connected rigid body system equations and their derivation, it seems appropriate here to add a few technical remarks that may contribute to that report's utility and to correct several significant errors in the text which have been brought to the author's attention.

- (1) In the last paragraph on page 30, the hinge torques τ^1 , τ^2 , and τ^3 are referred to. The symbol τ^j means $\tau^j g^j$.
- (2) On page 35, when considering the significance of Eq. (34), note that $C^T C = (\bar{C} + \hat{C})^T (\bar{C} + \hat{C}) = (U - \hat{C})(U + \hat{C}) \cong U + \dots$ (2nd-degree terms), so orthogonality is preserved in the linear approximation. Still, for planar motion (all \tilde{g} s the same), it would be impossible to associate *an* angle of rotation with C , since the cosine = 1 and the angle is large. Thus orthogonality is preserved, but physical interpretation is jeopardized.
- (3) Note the text corrections that follow.

Page	Line	Equation	Corrected Text	Comment
12	Last	—	$\epsilon_{\alpha\beta} V_\theta = \dots$	Change subscript
13	Last	—	the set $\mathcal{B} - \{k\}$ is \dots	Replace boldface k
14	17	—	$a_{0k} \stackrel{\Delta}{=} \sum_{r \in \mathcal{B}} \sum_{j \in \mathcal{B}} \epsilon_{kj} C^{0j} \Phi^{rj} C^{rk} g^k$	Change subscript
14	18	—	$-M \sum_{r \in \mathcal{B}} \sum_{j \in \mathcal{B} - k} \epsilon_{kj} (C^{0r} D^{jr}{}^T C^{jr} D^{ri} - \dots)$	Change subscript and summation index
15	12	—	$+ M \sum_{j \in \mathcal{B} - k} (\tilde{D}^{kj} C^{kj}) [\dots]$	Add tilde
23	4	23	$C = \begin{bmatrix} c_2 c_3 & -c_2 s_3 & s_2 \\ s_1 s_2 c_3 + s_3 c_1 & -s_1 s_2 s_3 + c_3 c_1 & c_1 s_2 s_3 + c_3 s_1 \\ -c_1 s_2 c_3 + s_3 s_1 & -s_1 c_2 & c_1 c_2 \end{bmatrix}$	Note several changes
33	1	30b	$+ \sum_{k \in \mathcal{P} - \mathcal{R}} (\bar{a}_{ik} + \hat{a}_{ik}) \ddot{\gamma}_k$	Change summation index
33	4th from last	—	Any terminal appendage in \dots	Drop s from appendages
35	26	35b	$+ \sum_{k \in \mathcal{R}} \bar{a}_{ik} \ddot{\gamma}_k + \sum_{k \in \mathcal{P} - \mathcal{R}} (\bar{a}_{ik} + \dots)$	Add subscript; change summation index
36	5	36b	$- M \sum_{r \in \mathcal{B}} \sum_{j \in \mathcal{B} - r} \epsilon_{kj} (\dots)$	Change subscript and summation index
36	19	37b	$- M \sum_{r \in \mathcal{B}} \sum_{j \in \mathcal{B} - r} \epsilon_{kj} [\dots]$	Change subscript
37	8	37e	$\dots [\bar{C}^{j0} \bar{w} \bar{C}^{0j} + \dots]$	Drop superscript j
37	11	37e	$\dots + \sum_{r \in \mathcal{P} - \mathcal{R}} \epsilon_{rj} \dot{\gamma}_r \bar{C}^{jr} \dots$	Add overdot
37	25	—	... coordinate truncation are feasible.	Replace "transformation is"
82	9	A-39a	$\sum_{k \in \mathcal{P}} \sum_{r \in \mathcal{B}} [\epsilon_{kr} \Phi^{rr} - M \sum_{j \in \mathcal{B} - r} \epsilon_{kj} (D^{ir} \cdot D^{rj} \mathbf{U} - D^{jr} D^{ri})] \cdot g^k \ddot{\gamma}_k = \dots$	Note several changes
82	25	A-40a	$g^s \cdot \{ \sum_{r \in \mathcal{P}} \sum_{i \in \mathcal{B}} [\epsilon_{ir} \Phi^{rr} - M \sum_{j \in \mathcal{B} - r} \epsilon_{ji} (D^{ir} \cdot D^{ri} \mathbf{U} - D^{jr} D^{ri})] \cdot g^i \ddot{\gamma}_j \} = \dots$	Note several changes
82	29	A-40a	$\dots + \sum_{s \in \mathcal{P}} \epsilon_{sr} g^s \dot{\gamma}_s) \times g^r \}$	Replace + by \times

Page	Line	Equation	Corrected Text	Comment
101	3	—	$[D^{01T}D^{10}U - D^{01}D^{10T}]$	Interchange superscripts
101	4	—	$- \gamma_1 D^{01T} \tilde{u}^1 D^{10} U - \gamma_1 D^{01T} D^{10} \tilde{u}^1 + D^{21T} D^{12} U - D^{21} D^{12T}$	Interchange superscripts
101	5	—	$- (\gamma_1 - \gamma_2) D^{21T} \tilde{u}^1 D^{12} U - \gamma_1 D^{21T} D^{12} \tilde{u}^1$	Interchange superscripts
101	9	—	$u^{1T} D^{01} D^{10} = u^{1T} D^{21} D^{12} = \dots$	Interchange superscripts
101	9	—	$D^{01T} u^1 D^{10} = D^{21T} \tilde{u}^1 D^{12} = 0$	Interchange superscripts

Contents

I. Introduction	1
II. Unrestricted Systems	3
A. System Mathematical Model	3
B. The Dynamical Equations	8
C. Kinematical Equations	10
D. System Specification	10
E. Subroutine MBDY Usage	11
F. A Five-Body System Simulation	13
III. Totally Linearized Systems	36
A. The Equations	36
B. Subroutines MBDYTL and MBDYL	64
C. Application of MBDYTL and MBDYL to the Five-Body System	65
IV. Partially Linearized Systems	79
A. The Equations	79
B. Subroutine MBDYPL Usage	80
C. Application of MBDYPL to the Five-Body System	81
V. Summary and Conclusions	82
Appendix A. Subroutine MBDY Description	99
Appendix B. Summary of MBDYTL, MBDYL User Requirements	108
Appendix C. Summary of MBDYPL User Requirements	120
References	130
 Figures	
1. Eleven rigid bodies interconnected by ten hinges	4
2. Alternate labeling schemes for a system of seven bodies	6
3. Augmented Δ_0 from Fig. 1	8
4. A space vehicle idealized as a system of five hinge-connected rigid bodies	15
5. Single-axis attitude control loop	16
6. Five-body system simulation using MBDY (Case 1)	18
7. Results of Case 1 simulation using MBDY	20

Contents (contd)

Figures (contd)

8. Results of Case 1 simulation using MBDY (components of ω^0 ; first 10 s)	30
9. Case 1 simulation program using MBDY, with prescribed platform motion	34
10. $\ddot{\gamma}_1$ prescribed function of time ($\ddot{\gamma}_2 = -\ddot{\gamma}_1$)	36
11. Results of Case 1 simulation with prescribed platform motion	37
12. Results of Case 1 simulation without control torques	47
13. Case 2 simulation program using MBDY	50
14. Results of Case 2 simulation	52
15. Case 2 simulation using MBDYTL	67
16. Results of Case 2 simulation using MBDYTL	69
17. Case 1 simulation program using MBDYPL	84
18. Results of Case 1 simulation using MBDYPL	86
19. Results of Case 1 simulation using MBDYPL (components of ω^0 ; first 10 s)	96

Abstract

This report describes several computer subroutines designed to provide the solution to minimum-dimension sets of discrete-coordinate equations of motion for systems consisting of an arbitrary number of hinge-connected rigid bodies assembled in a tree topology. In particular, these routines may be applied to (1) the case of completely unrestricted hinge rotations, (2) the totally linearized case (i.e., all system rotations are small), and (3) the mixed, or partially linearized, case. The use of the programs in each case is demonstrated using a five-body spacecraft and attitude control system configuration. The ability of the subroutines to accommodate prescribed motions of system bodies is also demonstrated. Complete listings and user instructions are included for these routines (written in Fortran V) which are intended as multi- and general-purpose tools in the simulation of spacecraft and other complex electromechanical systems.

Attitude Dynamics Simulation Subroutines for Systems of Hinge-Connected Rigid Bodies

I. Introduction

Since Hooker and Margulies (Ref. 1) published their derivation of the vector-dyadic equations of motion for an arbitrary number of point-connected rigid bodies in a topological tree, a number of computer programs (Refs. 2-7) based on these equations and similar generic formulations have come into wide use in the aerospace industry. Meanwhile, Hooker (Ref. 8) has devised a procedure for reducing the original Hooker-Margulies equations to a minimum-dimension set (i.e., equal to the number of degrees of rotational freedom) by summing selected subsets of the equations (paralleling Russell's approach—Ref. 2) and dot-multiplying by unit vectors associated with the connecting joint geometry.

In Reference (9) a variation of Hooker's procedure was used to derive a minimum-dimension set of equations of motion for a *hinge-connected* system of rigid bodies. (It is always possible to substitute a hinge-connected system for a point-connected system by introducing auxiliary bodies and without changing the dimension of the problem.) The objective was, having derived this set of *discrete-coordinate* equations in a highly systematic and detailed way, to (1) isolate certain subsets of the system's rigid bodies that may be considered collectively as a flexible substructure, (2) linearize in the variables describing relative motions of contiguous bodies within the substructure, and (3) transform from substructure discrete variables to distributed (modal) coordinates which may then be truncated for computational efficiency. The result is, in general, a new kind of *hybrid-coordinate* (discrete + distributed) formulation which can accommodate large gross deformations of an appendage even though its local deformations are small.

However, while computer subroutines which solve the hybrid-coordinate equations of Ref. 9 will be discussed in a subsequent report, this report will confine itself to descriptions of subroutines which solve only the discrete-coordinate equations of motion for the hinge-connected system. Specifically, it will describe and apply these programs as dynamic simulation tools for three possible situations: (1) the case in which all the system bodies are permitted large or *unrestricted* rotations (Section II), (2) the case in which all system rotations (and their derivatives) are small, allowing total linearization of the equations (Section III), and (3) the mixed case of large and small rotations for which partial linearization is appropriate (Section IV).

The programs, since they are based on the discrete-coordinate equations of Ref. 9, will differ from those developed in Ref. 4 in several significant respects. First, they will be of minimum dimension. Also, interbody coordinate transformations are accomplished internal to the programs, the user being required only to arbitrarily specify some nominal system configuration. Input has been simplified. Pains have been taken to minimize internal coordinate transformations and other computations to improve speed. And, finally, the user may, if he desires, *prescribe* the motion of some system bodies rather than allow the subroutine to calculate it.

The *subroutine* approach to solving attitude dynamics equations is a logical consequence of the approach taken historically at JPL to develop computer programs (analog or digital) specialized to each particular control mode or portion of a control mode under study, e.g., the thrust vector control mode during propulsive trajectory corrections. In contrast, however, to the narrow specialization of each simulation program to specific control problems has been the extensive use of *general-purpose simulation languages* to facilitate the creation of such programs. Over the past decade, such digital simulation languages as MIDAS, MIMIC, DSL/90, and most recently, CSSL III (Continuous System Simulation Language) have proved extremely valuable since they relieve the programmer of the worry over numerical integration routines, and since they can properly sort the many, at times highly interrelated, computational statements which occur and thereby help to ensure a correct solution.

Many of these "simulators" also provide a powerful array of useful special functions, in addition to the standard Fortran library of mathematical and intrinsic functions, such as hysteresis loops, noise generators, time delays, pulse generators, logic gates, etc. In fact, provision is also made for inclusion into the language of any *user-created* functions or subroutines. An example of an extremely useful subroutine that has been made a part of the CSSL III processor is HCK (Hamilton-Cayley-Klein), a rigid-body attitude dynamics package, written by E. H. Kopf. HCK calculates the angular acceleration components of a single rigid body, given its angular velocities, applied torques, and inertial properties. The HCK parameters (i.e., Euler parameters) are used in the necessary kinematic calculations and provide body-to-inertial (or vice versa) coordinate transformations as desired. Euler angles may also be requested to indicate the angular position of the body with respect to the inertial reference frame.

Just like HCK, the subroutines to be described here calculate angular accelerations, except that they do so for *every* body in a *system* of hinge-connected rigid bodies. All integration operations are confined to the main CSSL III simulation program (which calls its centralized numerical integration routines as required). Thus,

performing no numerical integration and required only to compute angular acceleration components (given the necessary system states, forcing functions, and physical properties), the routines are intended simply as very specialized and yet powerful simulation tools for application to a wide variety of current space vehicle attitude control problems.

II. Unrestricted Systems

A. System Mathematical Model

Figure 1 illustrates a set of eleven hinge-connected rigid bodies, labeled according to a convention designed to facilitate the processing of the equations of motion. This example will be useful in illustrating and interpreting the general mathematical model and the labeling conventions adopted here for its description. (In reading the list of conventions and symbol definitions, frequent reference should be made to Fig. 1.) Only after entertaining the complete cast of symbols used in what follows will consideration be given to the much smaller list of symbols to which values must be assigned by the user of the computer programs.

The mathematical model consists of a set of $n + 1$ rigid bodies interconnected by n hinges¹; the indicated numbers of bodies and hinges imply a tree topology. Any interbody connection forces in addition to those at the n hinges must be treated as forces external to the entire system. (If there were a linear spring connecting the mid-point of body \mathbf{b}_6 in Fig. 1 to some point on body \mathbf{b}_0 , for example, it would be necessary to replace the spring with a pair of equal and opposite forces applied to the spring attachment points on \mathbf{b}_6 and \mathbf{b}_0 ; the magnitude and direction of these forces would have to be expressed in terms of the unknown kinematical variables of the system.)

1. Definitions of fundamental symbols

Def. 1. Let n be the number of hinges interconnecting a set of $n + 1$ bodies.

Def. 2. Define the integer sets $\mathcal{B} \stackrel{\Delta}{=} \{0, 1, \dots, n\}$ and $\mathcal{P} \stackrel{\Delta}{=} \{1, \dots, n\}$.

Def. 3. Let \mathbf{b}_0 be a label assigned to one body chosen arbitrarily as a reference body, and let $\mathbf{b}_1, \dots, \mathbf{b}_n$ be labels assigned to the remaining bodies in such a way that if \mathbf{b}_j is located between \mathbf{b}_0 and \mathbf{b}_k , then $0 < j < k$.

Def. 4. Define the dextral orthogonal sets of unit vectors $\mathbf{b}_1^k, \mathbf{b}_2^k, \mathbf{b}_3^k$ so as to be imbedded in \mathbf{b}_k for $k \in \mathcal{B}$, and such that in some arbitrarily selected nominal configuration of the system $\mathbf{b}_{\alpha}^k = \mathbf{b}_{\alpha}'$ for $\alpha = 1, 2, 3$ and $k \in \mathcal{B}$.

Def. 5. Define $\{\mathbf{b}^k\}$ as the column array of unit vectors²

¹The word "hinge" as used here implies a connection which maintains a line common to both bodies; such a connection is sometimes called a *line hinge*.

²Note that braces {} are used in this report both to identify column arrays of vectors and to enclose sets of integers; the distinction is apparent as soon as the objects within the braces are identified as vectors (boldface) or scalars.

$$\{\mathbf{b}^k\} \triangleq \begin{Bmatrix} \mathbf{b}_1^k \\ \mathbf{b}_2^k \\ \mathbf{b}_3^k \end{Bmatrix}, \quad \text{for } k \in \mathcal{B}$$

Def. 6. Let $\omega^0 \triangleq \{\mathbf{b}^0\}^T \omega^0$ be the inertial angular velocity vector of \mathbf{e}_0 , so that ω^0 is the corresponding 3-by-1 matrix in vector basis $\{\mathbf{b}^0\}$.

Def. 7. Let c_k be the mass center of \mathbf{e}_k for $k \in \mathcal{B}$ (see Fig. 1).

Def. 8. Let \mathbf{f}_k be a point on the hinge axis common to \mathbf{e}_k and \mathbf{e}_j for $j < k$ and $k \in \mathcal{P}$ (see Fig. 1).

Def. 9. Define the k th neighbor set \mathcal{B}_k for $k \in \mathcal{B}$ such that $j \in \mathcal{B}_k$ if \mathbf{e}_j is attached to \mathbf{e}_k .

Def. 10. For $j \in \mathcal{B}_k$, let $\mathbf{p}^{kj} \triangleq \{\mathbf{b}^k\}^T \mathbf{p}^{kj}$ be the position vector with respect to c_k of the labeled point on the hinge axis (either \mathbf{f}_j or \mathbf{f}_k) which is common to \mathbf{e}_k and \mathbf{e}_j , so that \mathbf{p}^{kj} is the corresponding 3-by-1 matrix in vector basis $\{\mathbf{b}^k\}$. (See Fig. 1, and note the special cases $\mathbf{p}^{21} = \mathbf{p}^{28} = \mathbf{p}^{32} = \mathbf{p}^{54} = \mathbf{p}^{43} = 0$.)

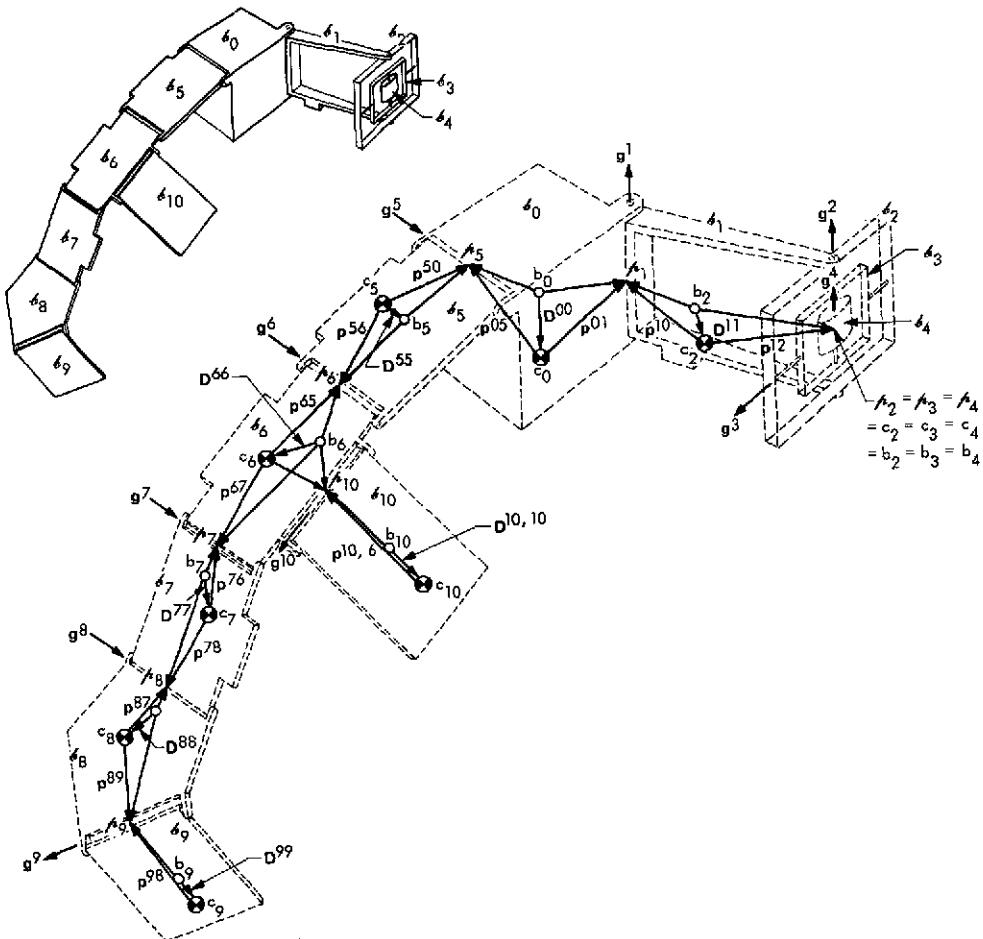


Fig. 1. Eleven rigid bodies interconnected by ten hinges

Def. 11. Let $\mathbf{g}^k \stackrel{\Delta}{=} \{\mathbf{b}^k\}^T \mathbf{g}^k$ be a unit vector parallel to the hinge axis through \mathbf{f}_k , so that \mathbf{g}^k is the corresponding 3-by-1 matrix in vector basis $\{\mathbf{b}^k\}$.

Def. 12. For $k \in \mathcal{P}$, let γ_k be the angle of a \mathbf{g}^k -rotation of \mathbf{f}_k with respect to the body attached at \mathbf{f}_k , that is, a rotation during which a right-handed screw fixed in \mathbf{f}_k with its axis parallel to \mathbf{g}^k advances in the direction of \mathbf{g}^k . Assign the value zero to γ_k for $k \in \mathcal{P}$ when the system is in its arbitrarily chosen nominal configuration, for which $\mathbf{b}_\alpha^k = \mathbf{b}_\alpha^j$, $\alpha = 1, 2, 3$; $k, j \in \mathcal{B}$.

Def. 13. Let m_k be the mass of \mathbf{f}_k for $k \in \mathcal{B}$.

Def. 14. Let $\mathbf{I}^k \stackrel{\Delta}{=} \{\mathbf{b}^k\}^T \mathbf{I}^k \{\mathbf{b}^k\}$ be the inertia dyadic of \mathbf{f}_k for c_k , so that \mathbf{I}^k is the corresponding inertia matrix in vector basis $\{\mathbf{b}^k\}$, i.e.,

$$\mathbf{I}^k = \begin{bmatrix} I_{11}^k & I_{12}^k & I_{13}^k \\ I_{12}^k & I_{22}^k & I_{23}^k \\ I_{13}^k & I_{23}^k & I_{33}^k \end{bmatrix}$$

Def. 15. Let $\mathbf{F}^k \stackrel{\Delta}{=} \{\mathbf{b}^k\}^T \mathbf{F}^k$ be the resultant vector of all forces applied to \mathbf{f}_k except for those due to interbody forces transmitted at hinge connections, so that \mathbf{F}^k is the corresponding 3-by-1 matrix in vector basis $\{\mathbf{b}^k\}$. (Thus, any interbody forces due to spring connections, etc., which are separate from the n hinge connections would contribute to \mathbf{F}^k , as would all forces from sources external to the system.)

Def. 16. Let $\mathbf{T}^k \stackrel{\Delta}{=} \{\mathbf{b}^k\}^T \mathbf{T}^k$ be the resultant moment vector with respect to c_k of all forces applied to \mathbf{f}_k except for those due to interbody forces transmitted at hinge connections, so that \mathbf{T}^k is the corresponding 3-by-1 matrix in vector basis $\{\mathbf{b}^k\}$.

Def. 17. Let τ_k be the scalar magnitude of the torque component applied to \mathbf{f}_k in the direction of \mathbf{g}^k by the body connected at \mathbf{f}_k .

Note that Def. 3 allows a certain freedom in labeling the system bodies after the reference body \mathbf{f}_0 has been chosen, the requirement always being that label values increase as one progresses outward from \mathbf{f}_0 along any chain of bodies. Figure 2 shows, for example, two equally acceptable ways of labeling a seven-body system.

Def. 18. Let the tilde symbol (\sim) signify in application to a 3-by-1 matrix \mathbf{V} with elements V_θ ($\theta = 1, 2, 3$) transformation to a skew-symmetric 3-by-3 matrix $\tilde{\mathbf{V}}$. Thus, the matrix $\tilde{\mathbf{V}}$ has the expanded form

$$\tilde{\mathbf{V}} \stackrel{\Delta}{=} \begin{bmatrix} 0 & -V_3 & V_2 \\ V_3 & 0 & -V_1 \\ -V_2 & V_1 & 0 \end{bmatrix}$$

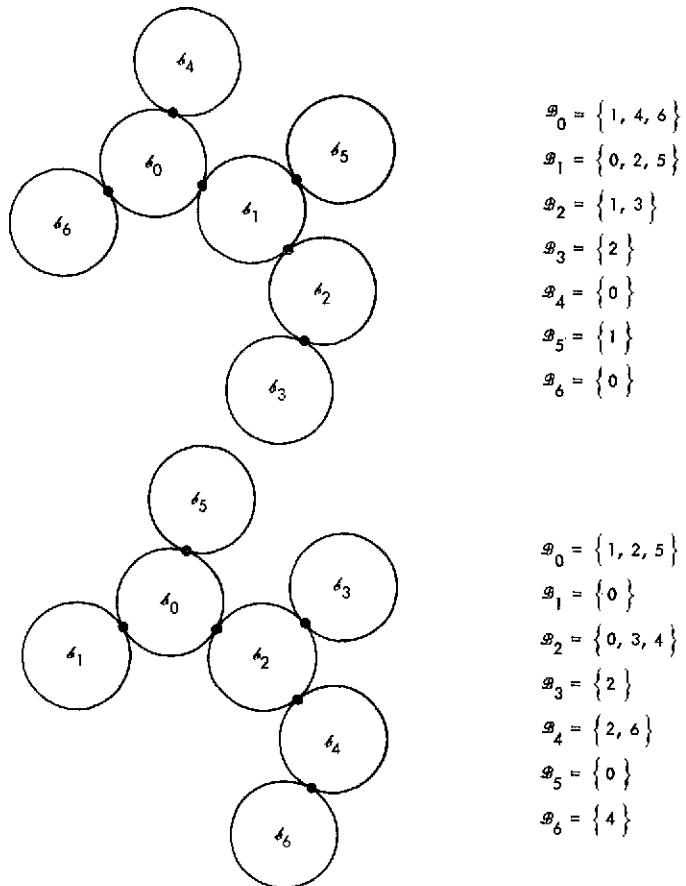


Fig. 2. Alternate labeling schemes for a system of seven bodies

2. Definitions of derived symbols

Def. 19. Define the scalar ε_{sk} such that for $k \in \mathcal{B}$ and $s \in \mathcal{P}$,

$$\varepsilon_{sk} \triangleq \begin{cases} 1 & \text{if } b_s \text{ lies between } b_0 \text{ and } b_k \\ 0 & \text{otherwise} \end{cases}$$

The term *path elements* will be applied to the scalars ε_{sk} , which are $n(n+1)$ in number.

Def. 20. Let C^{rj} be the direction cosine matrix defined by

$$\{\mathbf{b}^r\} \triangleq C^{rj} \{\mathbf{b}^j\}$$

for $r, j \in \mathcal{B}$. In the nominal state, $\gamma_k = 0$ for $k \in \mathcal{P}$, and all unit vectors of corresponding subscript are aligned, so that C^{rj} then is U , the 3-by-3 unit matrix.

Def. 21. Define $\mathcal{M} \triangleq \sum_{k \in \mathcal{B}} m_k$, so that \mathcal{M} is the total system mass.

Def. 22. Define $\mathcal{M}_{kj} \stackrel{\Delta}{=} \text{the total mass on the branch of the system attached to } \mathfrak{b}_k \text{ and commencing with } \mathfrak{b}_j$.

Def. 23. For $j \in \{\mathcal{B} - k\}$, define $\mathbf{L}^{kj} \stackrel{\Delta}{=} \text{the vector from } c_k \text{ to the hinge point attached to } \mathfrak{b}_k \text{ and on the path leading from } \mathfrak{b}_k \text{ toward } \mathfrak{b}_j$. (In Fig. 1, for example, $\mathbf{L}^{56} = \mathbf{L}^{57} = \mathbf{L}^{58} = \mathbf{p}^{56}$ and $\mathbf{L}^{3j} = 0$ for $j \in \mathcal{B}$.) (Define $\mathbf{L}^{kk} \stackrel{\Delta}{=} 0$.)

Def. 24. Define $\{\mathbf{b}^k\}^T \mathbf{D}^{kk} \stackrel{\Delta}{=} \mathbf{D}^{kk} \stackrel{\Delta}{=} -\sum_{j \in \mathcal{B}} \mathbf{L}^{kj} m_j / \mathcal{M}$ for $k \in \mathcal{B}$.

Def. 25. Let b_k be a point called the *barycenter* of \mathfrak{b}_k and fixed relative to \mathfrak{b}_k such that \mathbf{D}^{kk} is the position vector of c_k with respect to b_k .

Def. 26. Define $\{\mathbf{b}^k\}^T \mathbf{D}^{kj} \stackrel{\Delta}{=} \mathbf{D}^{kj} \stackrel{\Delta}{=} \mathbf{D}^{kk} + \mathbf{L}^{kj}$ for $k, j \in \mathcal{B}$. Thus, \mathbf{D}^{kj} is the vector from b_k to the hinge point fixed in \mathfrak{b}_k and on the path leading to \mathfrak{b}_j .

Def. 27. Define the dyadic

$$\Phi^{kk} \stackrel{\Delta}{=} \mathbf{I}^k + m_k (\mathbf{D}^{kk} \cdot \mathbf{D}^{kk} \mathbf{U} - \mathbf{D}^{kk} \mathbf{D}^{kk}) + \sum_{j \in \mathcal{B}_k} \mathcal{M}_{kj} (\mathbf{D}^{kj} \cdot \mathbf{D}^{kj} \mathbf{U} - \mathbf{D}^{kj} \mathbf{D}^{kj})$$

where \mathbf{U} is the unit dyadic.

Def. 28. For $j \in \{\mathcal{B} - k\}$, define the dyadic

$$\Phi^{kj} \stackrel{\Delta}{=} -\mathcal{M} (\mathbf{D}^{jk} \cdot \mathbf{D}^{kj} \mathbf{U} - \mathbf{D}^{jk} \mathbf{D}^{kj})$$

3. Augmented bodies and barycenters. The *barycenter* b_k of body \mathfrak{b}_k is defined in Def. 25 in mathematical terms which admit an interesting interpretation as a physical property of an abstraction called the *augmented body*. If one imagines the body \mathfrak{b}_k to be augmented by the addition of a particle of mass \mathcal{M}_{kj} at the connection point of \mathfrak{b}_k with \mathfrak{b}_j (for $j \in \mathcal{B}_k$), then the result is called the k th augmented body, here referred to as \mathfrak{b}'_k . Since \mathcal{M}_{kj} is the total mass of the branch of bodies attached to \mathfrak{b}_k commencing with \mathfrak{b}_j , the mass of *each* augmented body is \mathcal{M} , the total system mass.

Definition 24 provides

$$-\mathcal{M} \mathbf{D}^{kk} = \sum_{j \in \mathcal{B}} m_j \mathbf{L}^{kj}$$

which reveals the fact that $-\mathbf{D}^{kk}$ is the position vector of the mass center of \mathfrak{b}'_k with respect to the mass center c_k of b_k . Thus, *the barycenter* b_k is the mass center of the k th augmented body \mathfrak{b}'_k . With this interpretation comes the relationship

$$\sum_{j \in \mathcal{B}} m_j \mathbf{D}^{kj} = m_k \mathbf{D}^{kk} + \sum_{j \in \mathcal{B} - k} m_j \mathbf{D}^{kj} = 0$$

Since \mathbf{D}^{kk} is the position vector from the barycenter b_k to the mass center c_k of \mathfrak{b}_k , and (from Def. 26) \mathbf{D}^{kj} is the position vector from b_k to a particle of mass \mathcal{M}_{kj} located at the connection point of \mathfrak{b}_k leading to body \mathfrak{b}_j , for $j \in \mathcal{B}_k$, one can interpret Φ^{kk} from Def. 27 as the inertia dyadic of the augmented body \mathfrak{b}'_k with respect to the barycenter b_k . Such physical interpretation is not possible for the dyadic Φ^{kj} defined in Def. 28, since this dyadic involves vector \mathbf{D}^{jk} fixed in \mathfrak{b}_j and vector \mathbf{D}^{kj} fixed in \mathfrak{b}_k .

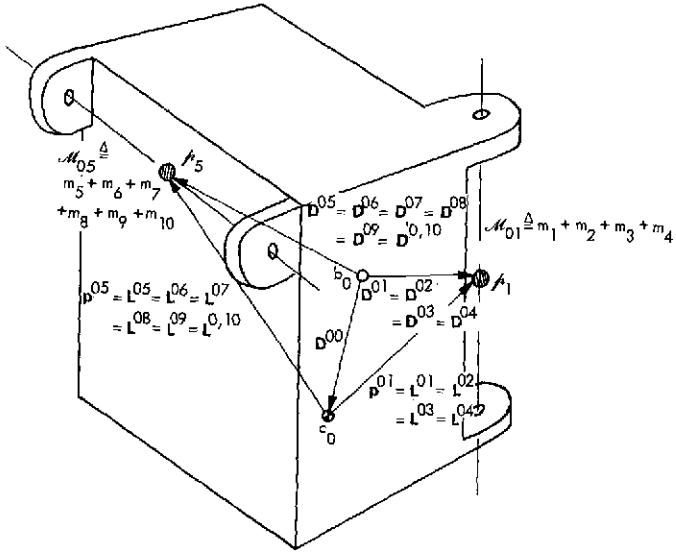


Fig. 3. Augmented \mathcal{B}_0 from Fig. 1

In illustration of the concepts just described, Fig. 3 shows the augmented body \mathcal{B}_0 associated with body \mathcal{B}_0 in the system of Fig. 1.

B. The Dynamical Equations

Using the symbols defined in the preceding section, the minimum-dimension set of equations for the rotational motion of the hinge-connected system (i.e., with n line hinges) can be written in vector-dyadic form as

$$\sum_{k \in \mathcal{B}} [T^k + \sum_{j \in \mathcal{B}} D^{kj} \times F^j - \Phi^{kk} \cdot \dot{\omega}^k - \omega^k \times \Phi^{kk} \cdot \omega^k + M \sum_{j \in \mathcal{B}-k} D^{kj} \times \ddot{D}^{jk}] = 0 \quad (1a)$$

$$\tau_r + g^r \cdot \sum_{k \in \mathcal{P}} \varepsilon_{rk} [T^k + \sum_{j \in \mathcal{B}} D^{kj} \times F^j - \Phi^{kk} \cdot \dot{\omega}^k - \omega^k \times \Phi^{kk} \cdot \omega^k + M \sum_{j \in \mathcal{B}-k} D^{kj} \times \ddot{D}^{jk}] = 0 \quad r \in \mathcal{P} \quad (1b)$$

This result is, of course, the same as that derived by Hooker (Ref. 8) and proved in Ref. 9. Equation (1b) provides n scalar equations, and the vector equation (1a) completes the minimum-dimension set of $n + 3$ scalar equations. By substituting the following kinematic expansions,

$$\ddot{D}^{jk} = \dot{\omega}^j \times D^{jk} + \omega^j \times (\omega^i \times D^{ik}) \quad (2)$$

and

$$\dot{\omega}^j = \dot{\omega}^0 + \sum_{s \in \mathcal{P}} \varepsilon_{sj} (g^s \gamma_s + \omega^s \times g^s \dot{\gamma}_s) \quad (3)$$

the vector-dyadic equations can be written in the form:

$$a^{00} \cdot \dot{\omega}^0 + \sum_{s \in \mathcal{P}} a^{ns} \gamma_s = \sum_{k \in \mathcal{B}} A^k \quad (4a)$$

and (for $r \in \mathcal{P}$)

$$a^{r0} \cdot \dot{\omega}^0 + \sum_{s \in \mathcal{P}} a_{rs} \ddot{\gamma}_s = g^r \cdot \sum_{k \in \mathcal{P}} \varepsilon_{rk} A^k + \tau_r \quad (4b)$$

where

$$Def. 29. \mathbf{a}^{00} \stackrel{\Delta}{=} \sum_{k \in \mathcal{B}} [\Phi^{kk} - M \sum_{j \in \mathcal{B}-k} (\mathbf{D}^{jk} \cdot \mathbf{D}^{kj} \mathbf{U} - \mathbf{D}^{ik} \mathbf{D}^{kj})] = \sum_{k \in \mathcal{B}} \sum_{j \in \mathcal{B}} \Phi^{kj}$$

$$\begin{aligned} Def. 30. \mathbf{a}^{0s} &\stackrel{\Delta}{=} \sum_{k \in \mathcal{B}} [\varepsilon_{sk} \Phi^{kk} - M \sum_{j \in \mathcal{B}-k} \varepsilon_{sj} (\mathbf{D}^{jk} \cdot \mathbf{D}^{kj} \mathbf{U} - \mathbf{D}^{ik} \mathbf{D}^{kj})] \cdot \mathbf{g}^s \\ &= \sum_{k \in \mathcal{B}} \sum_{j \in \mathcal{B}} \varepsilon_{sj} \Phi^{kj} \cdot \mathbf{g}^s \end{aligned}$$

$$\begin{aligned} Def. 31. \mathbf{a}^{r0} &\stackrel{\Delta}{=} \mathbf{g}^r \cdot \sum_{k \in \mathcal{P}} \varepsilon_{rk} [\Phi^{kk} - M \sum_{j \in \mathcal{B}-k} (\mathbf{D}^{jk} \cdot \mathbf{D}^{kj} \mathbf{U} - \mathbf{D}^{ik} \mathbf{D}^{kj})] \\ &= \mathbf{g}^r \cdot \sum_{k \in \mathcal{P}} \sum_{j \in \mathcal{B}} \varepsilon_{rk} \Phi^{kj} \end{aligned}$$

$$\begin{aligned} Def. 32. \mathbf{a}_{rs} &\stackrel{\Delta}{=} \mathbf{g}^r \cdot \sum_{k \in \mathcal{P}} \varepsilon_{rk} [\Phi^{kk} - M \sum_{j \in \mathcal{B}-k} (\mathbf{D}^{jk} \cdot \mathbf{D}^{kj} \mathbf{U} - \mathbf{D}^{ik} \mathbf{D}^{kj})] \cdot \varepsilon_{sj} \mathbf{g}^s \\ &= \mathbf{g}^r \cdot \sum_{k \in \mathcal{P}} \sum_{j \in \mathcal{B}} \varepsilon_{rk} \varepsilon_{sj} \Phi^{kj} \cdot \mathbf{g}^s \end{aligned}$$

$$\begin{aligned} Def. 33. \mathbf{A}^k &\stackrel{\Delta}{=} \mathbf{T}^k + \sum_{j \in \mathcal{B}} \mathbf{D}^{kj} \times \mathbf{F}^j - \boldsymbol{\omega}^k \times \boldsymbol{\Phi}^{kk} \times \boldsymbol{\omega}^k \\ &\quad + M \sum_{j \in \mathcal{B}-k} \{ \mathbf{D}^{kj} \times [\boldsymbol{\omega}^j \times (\boldsymbol{\omega}^j \times \mathbf{D}^{ik})] \} \\ &\quad - \sum_{j \in \mathcal{B}} \Phi^{kj} \cdot \sum_{s \in \mathcal{P}} \varepsilon_{sj} \dot{\gamma}_s (\boldsymbol{\omega}^s \times \mathbf{g}^s) \end{aligned}$$

A matrix equivalent to Eqs. (4a) and (4b) may be obtained by defining the following symbols:

Def. 34. Define the 3-by-3 matrix in vector basis $\{\mathbf{b}^0\}$ as $a_{00} \stackrel{\Delta}{=} \{\mathbf{b}^0\} \cdot \mathbf{a}^{00} \cdot \{\mathbf{b}^0\}^T$.

Def. 35. For $s \in \mathcal{P}$, define the 3-by-1 matrices in basis $\{\mathbf{b}^0\}$ as $a_{0s} \stackrel{\Delta}{=} \{\mathbf{b}^0\} \cdot \mathbf{a}^{0s}$, and let $a_{s0} \stackrel{\Delta}{=} a_{0s}^T$.

Def. 36. For $k \in \mathcal{B}$, define the 3-by-1 matrices in basis $\{\mathbf{b}^0\}$ as $E_k \stackrel{\Delta}{=} \{\mathbf{b}^0\} \cdot \mathbf{A}^k$.

In terms of these symbols, the system matrix equation can be written as

$$\left[\begin{array}{cccccc} a_{00} & a_{01} & a_{02} & \cdots & a_{0n} \\ a_{10} & a_{11} & a_{12} & \cdots & a_{1n} \\ a_{20} & a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{n0} & a_{n1} & a_{n2} & \cdots & a_{nn} \end{array} \right] \left[\begin{array}{c} \dot{\boldsymbol{\omega}}^0 \\ \dot{\boldsymbol{\gamma}}_1 \\ \dot{\boldsymbol{\gamma}}_2 \\ \vdots \\ \dot{\boldsymbol{\gamma}}_n \end{array} \right] = \left[\begin{array}{c} \sum_{k \in \mathcal{B}} E_k \\ g^{1T} C^{10} \sum_{k \in \mathcal{P}} \varepsilon_{1k} E_k + \tau_1 \\ g^{2T} C^{20} \sum_{k \in \mathcal{P}} \varepsilon_{2k} E_k + \tau_2 \\ \vdots \\ g^{nT} C^{n0} \sum_{k \in \mathcal{P}} \varepsilon_{nk} E_k + \tau_n \end{array} \right] \quad (5)$$

The coefficient matrix on the left side of (5) is a real, nonsingular, symmetric matrix which varies with time as the kinematical variables $\gamma_1, \dots, \gamma_n$ change in

value. In general, then, the solution to this set of differential equations requires (e.g., when using numerical integration) repeated solutions of the matrix equation, (i.e., at every integration step).

If one wishes to replace any variable γ_j ($j \in \mathcal{P}$) by a known function of time, then one can remove $\dot{\gamma}_j$ from the angular acceleration matrix on the left side of Eq. (5), removing also the corresponding row and column of its coefficient matrix, and then rewrite on the right side those terms just deleted by the removal of the column on the left. This will be performed automatically by the subroutines if the user desires.

Equation (5) constitutes a complete set of *rotational dynamics equations*, but they are not fully descriptive of the system motion until they are augmented by control equations specifying τ_1, \dots, τ_n and certain external forces and torques, and augmented also by kinematical equations as provided in the next section.

C. Kinematical Equations

The kinematical variables adopted in the preceding sections are as follows: γ_k for $k \in \mathcal{P}$ (Def. 12); C^{rj} for $r, j \in \mathcal{B}$ (Def. 20); and $\omega^0 \stackrel{\Delta}{=} \{\mathbf{b}^0\} \cdot \boldsymbol{\omega}^0$ (Def. 6). Although the equations of motion have been expressed in terms of these quantities, the latter are not all independent. Relationships among kinematical variables developed in this section must therefore either be considered in conjunction with the dynamical equations or be substituted into them to remove redundant variables whenever a solution is sought.

The direction cosine matrix C^{rj} (Def. 20) relates sets of orthogonal unit vectors fixed in \mathbf{b}_r and \mathbf{b}_j , and hence depends upon those angles γ_α for which \mathbf{f}_α lies between \mathbf{b}_r and \mathbf{b}_j , and also upon the corresponding unit vectors \mathbf{g}^α defining the intervening hinge axes. For the special case in which \mathbf{b}_r and \mathbf{b}_j are *contiguous* and $j < r$, it is always possible (see Ref. 9) to express C^{rj} (and C^{jr}) in terms of the single angle γ_r and the single matrix \mathbf{g}^r , as follows:

$$C^{rj} = U \cos \gamma_r - \tilde{g}^r \sin \gamma_r + \tilde{g}^r g^{rT} (1 - \cos \gamma_r) \quad (6)$$

and

$$C^{jr} = U \cos \gamma_r + \tilde{g}^r \sin \gamma_r + \tilde{g}^r g^{rT} (1 - \cos \gamma_r) = (C^{rj})^T \quad (7)$$

Definitions 34, 35, and 36 make clear that all $n + 3$ scalar equations in (5) are described in the vector basis $\{\mathbf{b}^0\}$ of the reference body. This is for the purpose of minimizing the number of coordinate transformation computations that must be performed by the subroutines. Thus, it becomes unnecessary to calculate C^{rj} for all $r \in \mathcal{B}, j \in \mathcal{B}$. It is only required that C^{rj} be determined where \mathbf{b}_r and \mathbf{b}_j are contiguous and, since $C^{0r} = C^{0j}C^{jr}$, to then derive matrices C^{0r} for $r \in \mathcal{P}$. An algorithm for accomplishing this task is described in Appendix A.

D. System Specification

Once the appropriate equations from the three preceding sections are embodied in a computer program, the user need not deal directly with all of the symbols and concepts introduced in the formulation of these equations. He can provide the computer with the very limited body of input required to specify his system, and then concentrate on interpreting the numerical integration output.

It is assumed that the engineer in his wisdom and experience has devised for his physical system a mathematical model consisting of a system of $n + 1$ bodies interconnected by n hinges (such as that illustrated in Fig. 1). He has labeled the bodies as indicated by Def. 3 (Section II A), and fixed in his mind a nominal configuration (for which $\gamma_k = 0$ for $k \in \mathcal{P}$ and $\mathbf{b}_\alpha^k = \mathbf{b}_\alpha^j$ for $j, k \in \mathcal{B}$ and $\alpha = 1, 2, 3$). One final aid, however, will be needed to communicate the system's interconnection pattern to the computer subroutine.

Def. 37. For $k \in \mathcal{P}$, let h_k denote the index of the body attached to \mathbf{b}_k and on the path leading to \mathbf{b}_0 . The scalars h_k will be termed "connection elements." (In Fig. 1, $h_1 = 0$, $h_2 = 1$, $h_3 = 2$, $h_4 = 3$, $h_5 = 0$, $h_6 = 5$, $h_7 = 6$, $h_8 = 7$, $h_9 = 8$, and $h_{10} = 6$.)

With the definition of h_k , it is now possible to specify a complete set of input information for the subroutine:

Subroutine input (required)

- (1) The integer n (see Def. 1).
- (2) The n connection elements h_k ($k \in \mathcal{P}$) (see Def. 37).
- (3) For $k \in \mathcal{B}$ and $j \in \mathcal{B}_k$, the 3-by-1 matrices p^{kj} (see Def. 10).
- (4) The n 3-by-1 matrices g^k for $k \in \mathcal{P}$ (see Def. 11).
- (5) The $n + 1$ masses m_k for $k \in \mathcal{B}$ (see Def. 13).
- (6) The $n + 1$ 3-by-3 inertia matrices I^k for $k \in \mathcal{B}$ (see Def. 14).
- (7) For $k \in \mathcal{B}$, the $2(n + 1)$ 3-by-1 matrices F^k and T^k (see Defs. 15 and 16).
- (8) For $k \in \mathcal{P}$, the n scalar values τ_k (see Def. 17).
- (9) Values for the $2n$ scalars γ_k and $\dot{\gamma}_k$ for $k \in \mathcal{P}$ (see Def. 12) and the three elements of the angular velocity matrix ω^0 (see Def. 6).

Subroutine input (optional)

Any kinematically prescribed variables, such as $\gamma_j(t)$ for some $j \in \mathcal{P}$. As noted in Section II B, it is possible to modify Eq. (5) so as to accommodate the substitution of an explicit function of time for a variable, such as $\gamma_j(t)$, which in Eq. (5) is treated as an unknown. If this option is elected, the computer program input must include the specification of variables to be prescribed and appropriate values for γ_j , $\dot{\gamma}_j$, and $\ddot{\gamma}_j$.

No additional input is required. The subroutine is programmed to construct numerically all intermediate concepts (such as the path elements ϵ_{sk} for $k \in \mathcal{B}$, $s \in \mathcal{P}$, the barycentric position matrices D^{kk} and D^{kj} for $k, j \in \mathcal{B}$, the augmented body inertia matrices Φ^{kk} , etc. It can then evaluate numerically the functions displayed in Eq. (5) and provide the solution for ω^0 and $\dot{\gamma}_1, \dots, \dot{\gamma}_n$. The subroutine and explicit details of its use are described in the following section.

E. Subroutine MBDY Usage

The computer subroutine, called MBDY, which provides the solution to Eq. (5), i.e., the solution to the unrestricted rotations of an arbitrary, hinge-connected, rigid-body tree system, may be exercised by means of either of two call statements.

The first of these initializes the program, supplying it with data which will remain constant throughout the dynamic simulation run.

Initializing Call Statement

CALL MBDY(NH,H,MB,MA,PB,PA,G,PI)

where

$NH =$ the integer $n =$ number of system hinges.

$H(k) =$ array containing the connection elements $h_k, k \in \mathcal{P}$
 $(H(1) = h_1, H(2) = h_2, \dots, H(n) = h_n)$

$MB(j) =$ array of reference body (\mathcal{B}_0) inertial constants; $j = 1, \dots, 7$

(Specifically: $MB(1) = I_{11}^0, MB(2) = I_{22}^0, MB(3) = I_{33}^0,$

$MB(4) = -I_{12}^0, MB(5) = -I_{13}^0, MB(6) = -I_{23}^0,$ and $MB(7) = m_0.$)

$MA(i,j) =$ array of appendage body ($\mathcal{B}_1, \dots, \mathcal{B}_n$) inertial constants; $i \in \mathcal{P}, j = 1, \dots, 7$ (Thus, $MA(i,1) = I_{11}^i, MA(i,2) = I_{22}^i, \dots, MA(i,7) = m_i.$)

$PB(i,j) =$ array containing elements of $p^{0i}; i \in \mathcal{B}_0, j = 1, 2, 3$

$PA(i,j,k) =$ array containing elements of $p^{ij}; i \in \mathcal{P}, j \in \mathcal{B}_i, k = 1, 2, 3$
(Exception!! If $j < i$, set $PA(i,j,k) = p^{ij}$. Example: $PA(3,3,1) = p_1^{32}$. All $PA(i,j,k)$, where $j < i$, will be ignored.)

$G(i,j) =$ array containing elements of $g^i; i \in \mathcal{P}, j = 1, 2, 3$

$PI(i) =$ array of prescribed variable indicators; $i \in \mathcal{P}$ (If γ_i is a prescribed variable, $PI(i) = 1$. Otherwise, $PI(i) = 0.$)

The statement **CALL MBDY(NH,H,...)** need only be executed *once* prior to a simulation run, but subsequently the subroutine must be entered (at every integration step) to compute the angular accelerations $\ddot{\omega}^0, \ddot{\gamma}_1, \dots, \ddot{\gamma}_n$, i.e., the system's dynamic response to its forcing functions. This is accomplished by executing the "dynamic" call statement.

Dynamic Call Statement

CALL MRATE(NH,TH,TB,TA,FB,FA,GM,GMD,GMDD,WO,WDOT)

where

$NH =$ the integer $n =$ number of system hinges

$TH(i) =$ array containing the hinge torques, $\tau_i; i \in \mathcal{P}$

$TB(j) =$ array containing the elements of $T^0; j = 1, 2, 3$

$TA(i,j) =$ array containing the elements of $T^i; i \in \mathcal{P}, j = 1, 2, 3$

$FB(j) =$ array containing the elements of $F^0; j = 1, 2, 3$

$FA(i,j) =$ array containing the elements of $F^i; i \in \mathcal{P}, j = 1, 2, 3$

$GM(i) =$ array of the angles, $\gamma_i; i \in \mathcal{P}$

$GMD(i) =$ array of the angular velocities, $\dot{\gamma}_i; i \in \mathcal{P}$

$GMDD(i) =$ array of the *prescribed* angular accelerations, $\ddot{\gamma}_i; i \in \mathcal{P}$

$WO(j)$ = array containing the components of ω^0 ; $j = 1, 2, 3$

$WDOT(j)$ = solution vector containing the elements of $\dot{\omega}^0$ and $\ddot{\gamma}_1, \dots, \ddot{\gamma}_n$;
 $j = 1, \dots, n + 3$ ($WDOT(1) = \dot{\omega}_1^0$, $WDOT(2) = \dot{\omega}_2^0$, $WDOT(3) = \dot{\omega}_3^0$, $WDOT(4) = \ddot{\gamma}_1$, $WDOT(5) = \ddot{\gamma}_2, \dots, WDOT(n + 3) = \ddot{\gamma}_n$)

The solution vector array, $WDOT$, is returned to the main calling program by the subroutine and contains the current computed value for the reference body's angular acceleration relative to the inertial frame ω^0 and the relative angular accelerations $\ddot{\gamma}_i$ of the remaining appendage bodies.

To summarize, the call to $MRATE$ supplies the subroutine with current instantaneous values for hinge torques and externally applied torques and forces on the system. Explicit expressions for computing the T^i , F^i , and τ_i , often as functions of the γ_i and $\dot{\gamma}_i$ and other system or control variables, are located in the main simulation program (see sample problem which follows). Current values of ω^0 , γ_i , and $\dot{\gamma}_i$ are continuously produced by the main program's numerical integration operators (operating on the solution vector components, $WDOT(j)$) and are therefore always available for input to the subroutine. The $\ddot{\gamma}_i$ inputs to the subroutine are used only when the motion of certain system bodies is prescribed.

When the $MBDY$ subroutine is used, the main calling program must contain Fortran V (or IV) statements which specify "type" and allocate storage for the variables and arrays being used and assign values to these variables and arrays. Some examples of this will be shown in the next section. However, the *mandatory* specification statements are listed here.

Required Specification Statements

INTEGER NH, H(n), PI(n)

REAL MB(7), MA($n, 7$), PB($n, 3$), PA($n, n, 3$), G($n, 3$), TH(n),
TB(3), TA($n, 3$), FB(3), FA($n, 3$), GM(n), GMD(n),
GMDD(n), WO(3)

DOUBLE PRECISION WDOT($n + 3$)

Also, in order that storage allocation for arrays internal to the subroutine $MBDY$ be minimized (i.e., made consistent with n hinges and no more), the following Fortran V (or IV) statement must appear *in the subroutine*:

PARAMETER Q = n

The proper placement of this statement in $MBDY$ is shown in the subroutine listing (Appendix A).

F. A Five-Body System Simulation

To illustrate the application of the $MBDY$ subroutine for the unrestricted motion case, the space vehicle configuration shown in Fig. 4 will be simulated. The vehicle

is idealized to consist of a rigid central body, which will be designated as the reference body \mathbf{b}_0 , including the bus, solar panels, gas jets, sensors, communications equipment, etc.

Hinged to the reference body is a rotating hub, labeled \mathbf{b}_1 , which in turn is hinged to platform \mathbf{b}_2 , carrying a television camera. The platform hinge axes located by \mathbf{g}^1 and \mathbf{g}^2 are orthogonal, $\mathbf{g}^1 = \mathbf{b}_3^0$ (+Z-direction), and the rotation angles γ_1 and γ_2 are arbitrarily defined to be zero when the TV boresight unit vector \mathbf{u} is in the Y-direction. Also, \mathbf{u} is defined perpendicular to \mathbf{g}^2 . Thus, we may write

$$\begin{aligned}\mathbf{u} &= -\sin \gamma_1 \cos \gamma_2 \mathbf{b}_1^0 + \cos \gamma_1 \cos \gamma_2 \mathbf{b}_2^0 - \sin \gamma_2 \mathbf{b}_3^0 \\ \mathbf{g}^1 &= \mathbf{b}_3^1 \\ \mathbf{g}^2 &= -\mathbf{b}_1^2\end{aligned}$$

A large instrumented boom, \mathbf{b}_4 , is also hinged to the reference body by the mutually orthogonal and coplanar axes defined by \mathbf{g}^3 and \mathbf{g}^4 . (In this case, the intermediate rigid body, \mathbf{b}_3 , will be assigned zero mass.) Therefore,

$$\begin{aligned}\mathbf{g}^3 &= \mathbf{b}_1^3 \\ \mathbf{g}^4 &= \mathbf{b}_3^4\end{aligned}$$

The following list of system parameters will be used to define the system and initialize the MBDY subroutine:

- (1) $n = 4$
- (2) $h_1 = 0, h_2 = 1, h_3 = 0, h_4 = 3$
- (3) $p^{01} = [0 \ 0 \ -1.5]^T, p^{02} = [0 \ -1.20 \ 0]^T$ meters
 $p^{10} = [0 \ 0 \ -0.75]^T, p^{12} = [0 \ -0.10 \ -0.75]^T$ meters
 $p^{21} = [0 \ -0.22 \ 0.20]^T$ meters
 $p^{30} = [0 \ 0 \ 0]^T, p^{34} = [0 \ 0 \ 0]^T$ meters
 $p^{43} = [0 \ 3.3 \ 0]^T$ meters
- (4) $\mathbf{g}^1 = [0 \ 0 \ 1]^T$ meters
 $\mathbf{g}^2 = [-1 \ 0 \ 0]^T$ meters
 $\mathbf{g}^3 = [1 \ 0 \ 0]^T$ meters
 $\mathbf{g}^4 = [0 \ 0 \ 1]^T$ meters
- (5) $m_0 = 410$ kg
 $m_1 = 6.8$ kg
 $m_2 = 57.5$ kg
 $m_3 = 0$
 $m_4 = 10.7$ kg

$$(6) \quad I^0 = \begin{bmatrix} 115 & -14 & 14 \\ & 316 & -34.6 \\ \text{sym.} & & 440 \end{bmatrix}, \quad I^1 = \begin{bmatrix} 0.35 & 0 & 0 \\ & 0.35 & 0 \\ \text{sym.} & & 0 \end{bmatrix} \text{ kg-m}^2$$

$$I^2 = \begin{bmatrix} 4.85 & 0.41 & -0.07 \\ & 2.20 & 0.54 \\ \text{sym.} & & 5.50 \end{bmatrix}, \quad I^3 = \begin{bmatrix} 0 & 0 & 0 \\ & 0 & 0 \\ \text{sym.} & & 0 \end{bmatrix} \text{ kg-m}^2$$

$$I^4 = \begin{bmatrix} 27.2 & 0 & 0 \\ & 0.2 & 0 \\ \text{sym.} & & 27.2 \end{bmatrix} \text{ kg-m}^2$$

This completes the list of constants that are needed to characterize the system. It remains then to develop expressions for the dynamic forcing functions T^k , F^k , and τ_k .

Control of the vehicle shown in Fig. 4 is to be maintained by firing pairs of gas jets located on the reference body to produce, ideally, a pure torque couple on that body. It will be assumed in the simulations to follow that this is the case and that no net force is applied to any of the system bodies; i.e., $F^k = 0$ for $k \in \mathcal{P}$. The control torque T^0 can be expressed by

$$T^0 = [L_1 f_1 \quad L_2 f_2 \quad L_3 f_3]^T$$

where the control functions f_1 , f_2 , and f_3 each may take on the values ± 1 or 0, depending on the single-axis control law defined in Fig. 5. (All other T^k for $k \in \mathcal{P}$ are assumed to be zero.) The control law is based on position feedback signals derived from celestial sensors (i.e., sun sensors, star trackers) as well as rate feedback derived from gyro measurements. If the sum of these feedback errors is

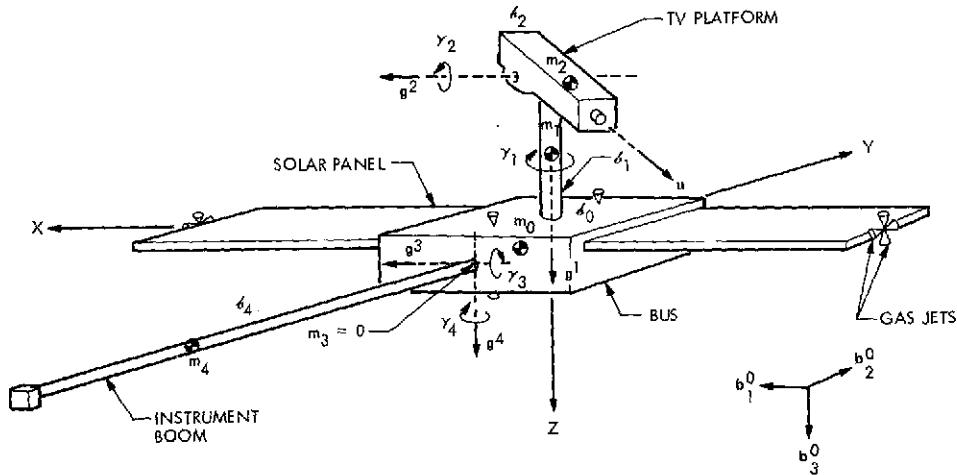


Fig. 4. A space vehicle idealized as a system of five hinge-connected rigid bodies

greater than some arbitrary value ($\pm\theta_{DB}$ in Fig. 5), a switching amplifier fires the gas jets to apply corrective torques. A list of values assigned to the reference body attitude control parameters follows:

$$\begin{aligned}\theta_{DB} &= 0.005 \text{ rad} & L_1 &= 0.23 \text{ n-m} \\ \tau_R &= 2.0 \text{ s} & L_2 &= 0.21 \text{ n-m} \\ & & L_3 &= 0.31 \text{ n-m} \\ & & TM &= 0.020 \text{ s}\end{aligned}$$

Finally, it is necessary to characterize the hinge torques $\tau_k (k \in \mathcal{P})$. Since the boom will not be articulated, the boom hinge torques can be described by

$$\begin{aligned}\tau_3 &= -K_3\gamma_3 - B_3\dot{\gamma}_3 \\ \tau_4 &= -K_4\gamma_4 - B_4\dot{\gamma}_4\end{aligned}$$

where

$$\begin{aligned}K_3 &= 2000 \text{ n-m/rad}, & B_3 &= 10 \text{ n-m-s/rad} \\ K_4 &= 2000 \text{ n-m/rad}, & B_4 &= 10 \text{ n-m-s/rad}\end{aligned}$$

However, since the TV platform will be articulated relative to its nominal pointing direction, it is convenient to introduce the variables γ_{1c} and γ_{2c} to represent the commanded position of the platform. The hinge torques can then be written as

$$\begin{aligned}\tau_1 &= -K_1(\gamma_1 - \gamma_{1c}) - B_1\dot{\gamma}_1 \\ \tau_2 &= -K_2(\gamma_2 - \gamma_{2c}) - B_2\dot{\gamma}_2\end{aligned}$$

where

$$\begin{aligned}K_1 &= 3500 \text{ n-m/rad}, & B_1 &= 20 \text{ n-m-s/rad} \\ K_2 &= 3500 \text{ n-m/rad}, & B_2 &= 20 \text{ n-m-s/rad}\end{aligned}$$

1. Case 1—large platform maneuver. The first simulation of the five-body system of Fig. 4 and its controller will show its dynamic response to a large platform

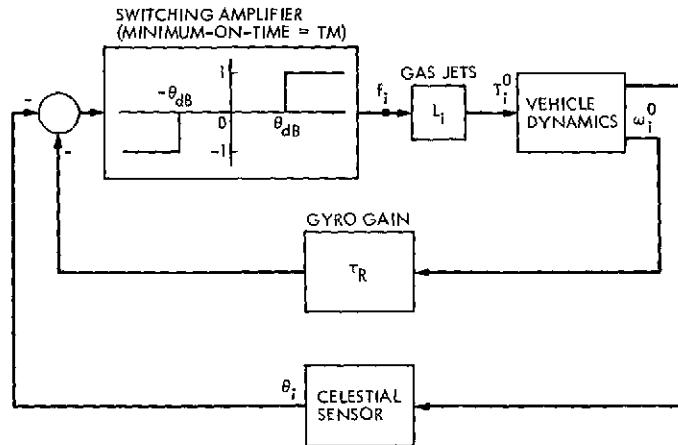


Fig. 5. Single-axis attitude control loop

maneuver. Initially, the platform is positioned at $\gamma_1 = 218$ deg and $\gamma_2 = -30$ deg. All other system variables, i.e., angular positions and rates, are assumed initially at zero. At $t = 0$, the command is issued to rotate the platform (about both axes simultaneously) at the constant rate of 1 deg/s to the position $\gamma_1 = 168$ deg, $\gamma_2 = 20$ deg. The program used to simulate this maneuver is shown in Fig. 6.

Note that the required MBDY specification statement REAL MB(7),... has been replaced by its equivalent in the CSSL III language, i.e., ARRAY MB(7),...

Also, note that the HCK routine mentioned in Section I is used to establish the position of the reference body in the inertially fixed frame, and it thereby provides a means for calculating θ_1 , θ_2 , and θ_3 , the attitude angles measured by celestial sensors fixed to the reference body.

Three other CSSL III built-in functions are also used in the simulation. The function "SWAMP" simulates the switching amplifier shown in Fig. 5. "STEP" provides the unit step function when the independent variable, TIME, is greater than the specified constant. And finally, "INTEG(a_1, a_2)" signifies the integration operation, where a_1 is the integrand and a_2 is the initial condition.

All arithmetic statements are in Fortran, although CSSL III allows several statements on a single line if separated by a ". Variables to be plotted at every communication interval CI are listed in the PREPAR statement. Printed variables are listed in the OUTPUT statement.

The statement "CALL MBDY(NH,...)" is located in the INITIAL section and is therefore executed only once, i.e., prior to the dynamic calculations. However, "CALL MRATE(NH,...)" is in the DERIVATIVE section and is thus executed at every integration step, as required.

Results of the simulation described in Fig. 6 are shown in the computer plots of Fig. 7. All variables plotted are in terms of radians, radians per second, and seconds.

Notice that all three components (W_1 , W_2 , W_3) of the reference body angular velocity ω^o , show evidence of platform hinge vibrations immediately following platform acceleration or deceleration (i.e., at $t = 0$ and $t = 50$ s). Only W_1 and W_3 show any significant evidence of the lower-frequency boom vibrations, since the boom is nominally aligned with the reference body's "2" axis. Attitude angles θ_1 , θ_2 , and θ_3 (THET1, THET2, THET3) have obviously been effectively confined to the ± 0.005 -rad deadband by the control system. Platform and boom angular rate and position responses clearly show the transients at $t = 0$ and $t = 50$ s. Some platform vibration is discernible in the boom responses.

Figure 8 repeats the transient solutions for W_1 , W_2 , and W_3 over just the first 10 s in order to display the response in better detail. The W_2 response indicates that control jets began pulsing at about $t = 4.6$ s in that axis, while pulsing in the other axes was not required until much later.

2. Case 1 with prescribed motion. In order to demonstrate the utility of the MBDY subroutine's option for prescribed motion, Case 1 was rerun with the platform rotations γ_1 and γ_2 *prescribed*. As indicated in Section II E, the motion of any

```

PROGRAM 5-BODY SYSTEM SIMULATION USING MBDY --- CASE 1
  INTEGER NH,PI(4),H(4)
  ARRAY MB(7),MA(4,7),PB(4,3),PA(4,4,3),G(4,3),TB(3),TA(4,3)
  ARRAY FB(3),FA(4,3),TH(4),GM(4),GMD(4),GMDD(4),WD(3)
  DOUBLE PRECISION WDOT(7)
  DATA H/0.,1.,0.,3./PI/0.,0.,0./
  DATA MB/115.,316.,440.,14.,-14.,34.,6.,410./
  DATA MA(1,1)/.35/MA(1,2)/.35/MA(1,7)/6.*8/
  DATA MA(2,1)/4.85/MA(2,2)/2.*2/MA(2,3)/5.5/MA(2,4)/-.41/MA(2,5)/.177/
  DATA MA(2,6)/-.54/MA(2,7)/57.5/
  DATA MA(4,1)/27.*2/MA(4,2)/.2/MA(4,3)/27.*2/MA(4,7)/10.7/
  DATA PB(1,3)/-1.5/PB(3,2)/-1.2/
  DATA PA(1,1,31)/-.75/PA(1,2,2)/-.10/PA(1,2,3)/-.75/
  DATA PA(2,2,2)/-.22/PA(2,2,3)/.20/
  DATA PA(4,4,2)/3.3/
  DATA G(1,3)/1./G(2,1)/-1./G(3,1)/1./G(4,3)/1./
  CONSTANT TM=.020,DB=.005,TR=2.0
  CONSTANT TFINAL=60.,CLKTIM=1100.
  CONSTANT L1=.23,L2=.21,L3=.31
  CONSTANT SLRAT=.01745329
  CONSTANT PHIZ=0.,THETZ=0.,PSIZ=0.
  CONSTANT W1=0.,W2=0.,W3=0.
  CONSTANT K1=3500.,K2=3500.,K3=2000.,K4=2000.
  CONSTANT B1=20.,B2=20.,B3=10.,B4=10.
  CONSTANT IC1=218.,IC2=-30.

INITIAL
  NH=4
  RD=57.2957795 $ DR=1./RD $ GM1C1=IC1/RD $ GM2C1=IC2/RD
  GM1=IC1/RD $ GM2=IC2/RD
  CALL MBDY(NH,H,MB,MA,PB,PA,G,PI)
  PZI,PA1,PB1,PC1=HCK(INITL,PHIZ,THETZ,PSIZ)

END
DYNAMIC
  IF(TIME>TFINAL) GO TO S1
DERIVATIVE
  BODY'S
    VARIABLE TIME=0.
    CINTERVAL CI=.06
    XERROR W1=1.E-6,W2=1.E-6,W3=1.E-6,GM3=1.E-4
    STPCLK CLKTIM
    OUTPUT 50,W1D,W2D,W3D,W1,W2,W3,THET1,THET2,THET3,..
      IN1,IN2,IN3,GM1,GM2,GM3,GM4,PL1,PL2
    PREPAR W1,W2,W3,THET1,THET2,THET3,GM1D,GM2D,..
      GM1,GM2,GM3D,GM4D,GM3,GM4
NOSORT
  W0(1)=W1 $ W0(2)=W2 $ W0(3)=W3
  PL1=RD*GM1 $ PL2=RD*GM2
  GM(1)=GM1 $ GM(2)=GM2 $ GM(3)=GM3 $ GM(4)=GM4
  GMD(1)=GM1D $ GMD(2)=GM2D $ GMD(3)=GM3D $ GMD(4)=GM4D
COMMENT
COMMENT CELESTIAL-SENSOR-MEASURED ANGLES
COMMENT
  D1,D2=HCK(MATRIX,PZI,PAQ,PBQ,PC0)
  NX0,NY0,NZ0=HCK(1TOB,0.,0.,1.,D1,D2)
  LX0,LY0,LZ0=HCK(1TOB,1.,0.,0.,D1,D2)
  THET1=ATAN2(NY0,NZ0) $ THET2=ATAN2(-NX0,NZ0)
  THET3=ATAN2(-LY0,LX0)

```

Fig. 6. Five-body system simulation using MBDY (Case 1)

```

COMMENT           RATE FEEDBACK SIGNAL AND SWITCHING AMP. DYNAMICS
COMMENT
COMMENT           IN1=-THET1-TR*#1   $  IN2=-THET2-TR*#2   $  IN3=-THET3-TR*#3
IN1=-THET1-TR*#1   $  IN2=-THET2-TR*#2   $  IN3=-THET3-TR*#3
AMP1=SWAMP(TIME,TM,DB,IN1)
AMP2=SWAMP(TIME,TM,DB,IN2)
AMP3=SWAMP(TIME,TM,DB,IN3)

COMMENT           PLATFORM POSITION COMMANDS
COMMENT
COMMENT           GM1CD=SLRAT*(-1.+STEP(50.,TIME)) $ GM1C=INTEG(GM1Cn,GM1C)
GM1CD=SLRAT*(-1.+STEP(50.,TIME)) $ GM1C=INTEG(GM1Cn,GM1C)
GM2CD=SLRAT*(1.-STEP(50.,TIME)) $ GM2C=INTEG(GM2Cn,GM2C)
COMMENT           REFERENCE BODY EXTERNAL TORQUE
COMMENT           HINGE TORQUES
COMMENT
TB(1)=L1*AMP1   $  TB(2)=L2*AMP2   $  TB(3)=L3*AMP3
TH(1)=-K1*(GM1-GM1C)-B1*GM1D
TH(2)=-K2*(GM2-GM2C)-B2*GM2D
TH(3)=-K3*GM3-B3*GM3D
TH(4)=-K4*GM4-B4*GM4D

COMMENT           SYSTEM ANGULAR ACCELERATIONS
COMMENT
CALL MRATE(NH,TH,TB,TA,FB,FA,GM,GMD,GMDD,W0,WDOT)
W1D=WDOT(1) $ W2D=WDOT(2) $ W3D=WDOT(3)

COMMENT           SYSTEM ANGULAR RATES AND POSITIONS
COMMENT
W1=INTEG(WDOT(1),W1)
W2=INTEG(WDOT(2),W2)
W3=INTEG(WDOT(3),W3)
GM1D=INTEG(WDOT(4),0.)      $  GM2D=INTEG(WDOT(5),0.)
GM3D=INTEG(WDOT(6),0.)      $  GM4D=INTEG(WDOT(7),0.)
GM1=INTEG(GM1D,GM1I)       $  GM2=INTEG(GM2D,GM2I)
GM3=INTEG(GM3D,0.)         $  GM4=INTEG(GM4D,0.)

COMMENT           HCK PARAMETER RATES AND POSITIONS
COMMENT
PZOD,PAOD,PBOD,PCOD=HCK(HCK,PZO,PAO,PBO,PCO,W1,W2,W3)
PZO=INTEG(PZOD,PZI)
PAO=INTEG(PAOD,PAI)
PBO=INTEG(PBOD,PBI)
PCO=INTEG(PCOD,PCI)

END
END
END
TERMINAL
SI** CONTINUE
END
END

```

Fig. 6 (contd)

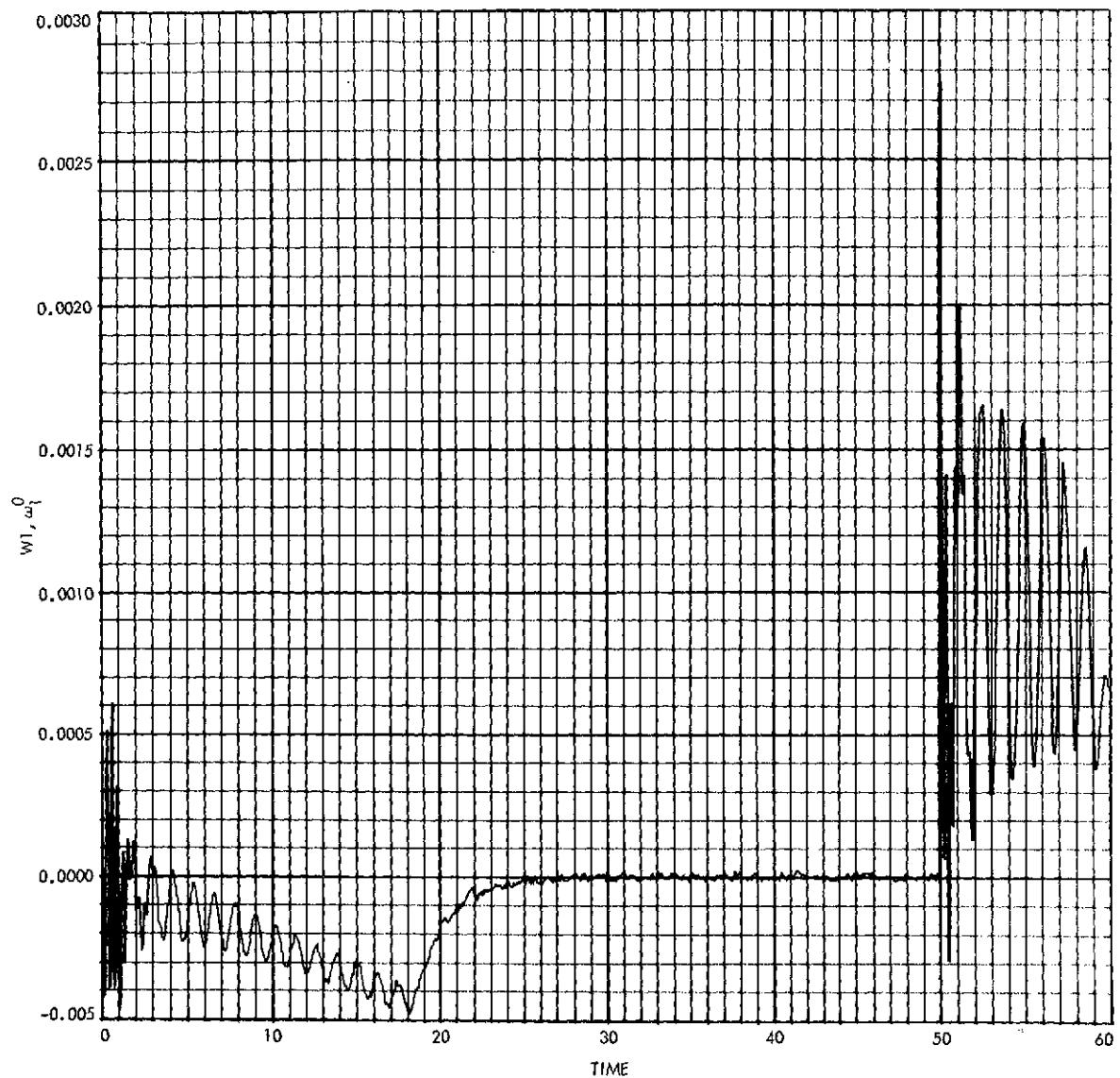


Fig. 7. Results of Case 1 simulation using MBDY

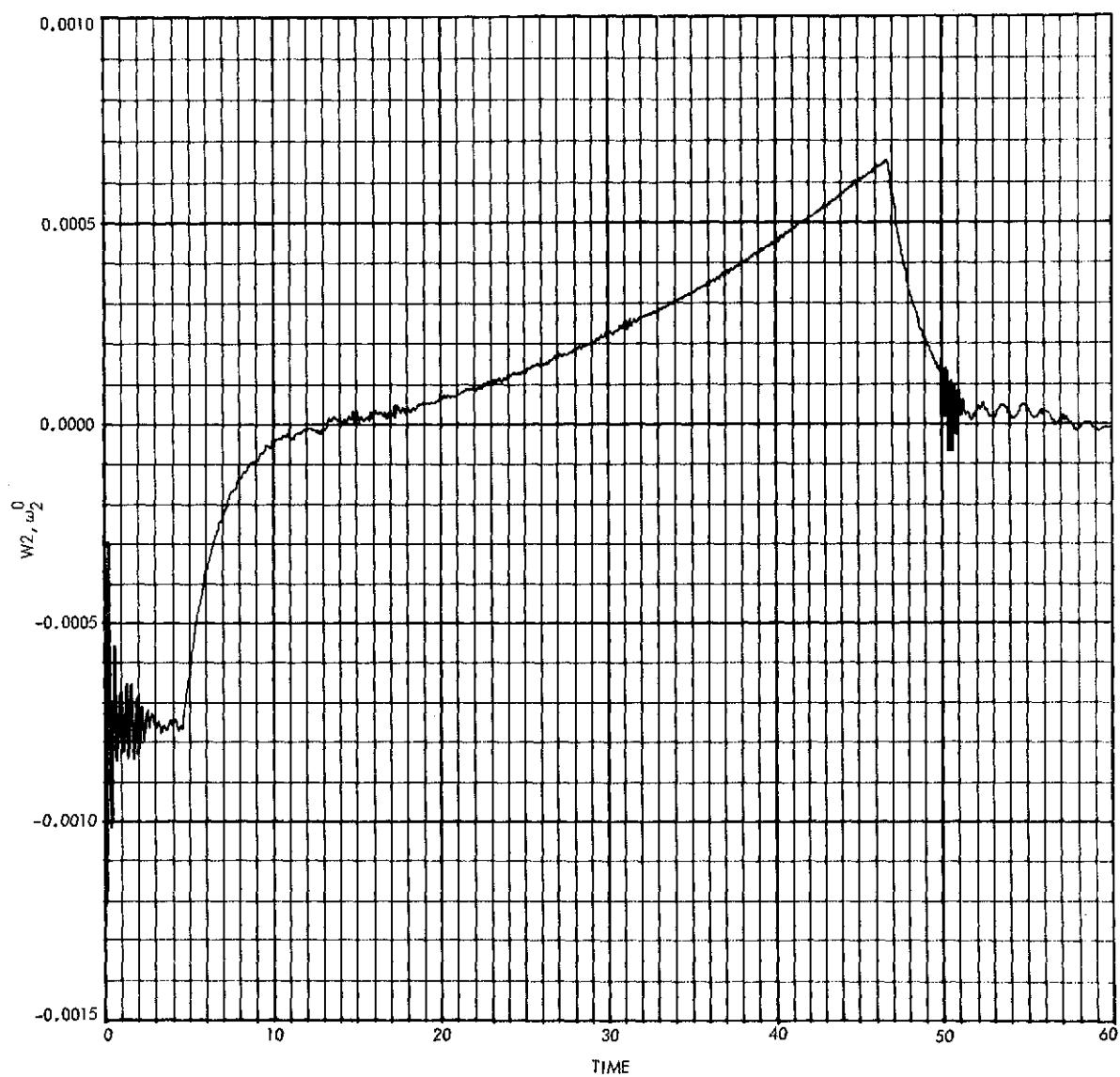


Fig. 7 (contd)

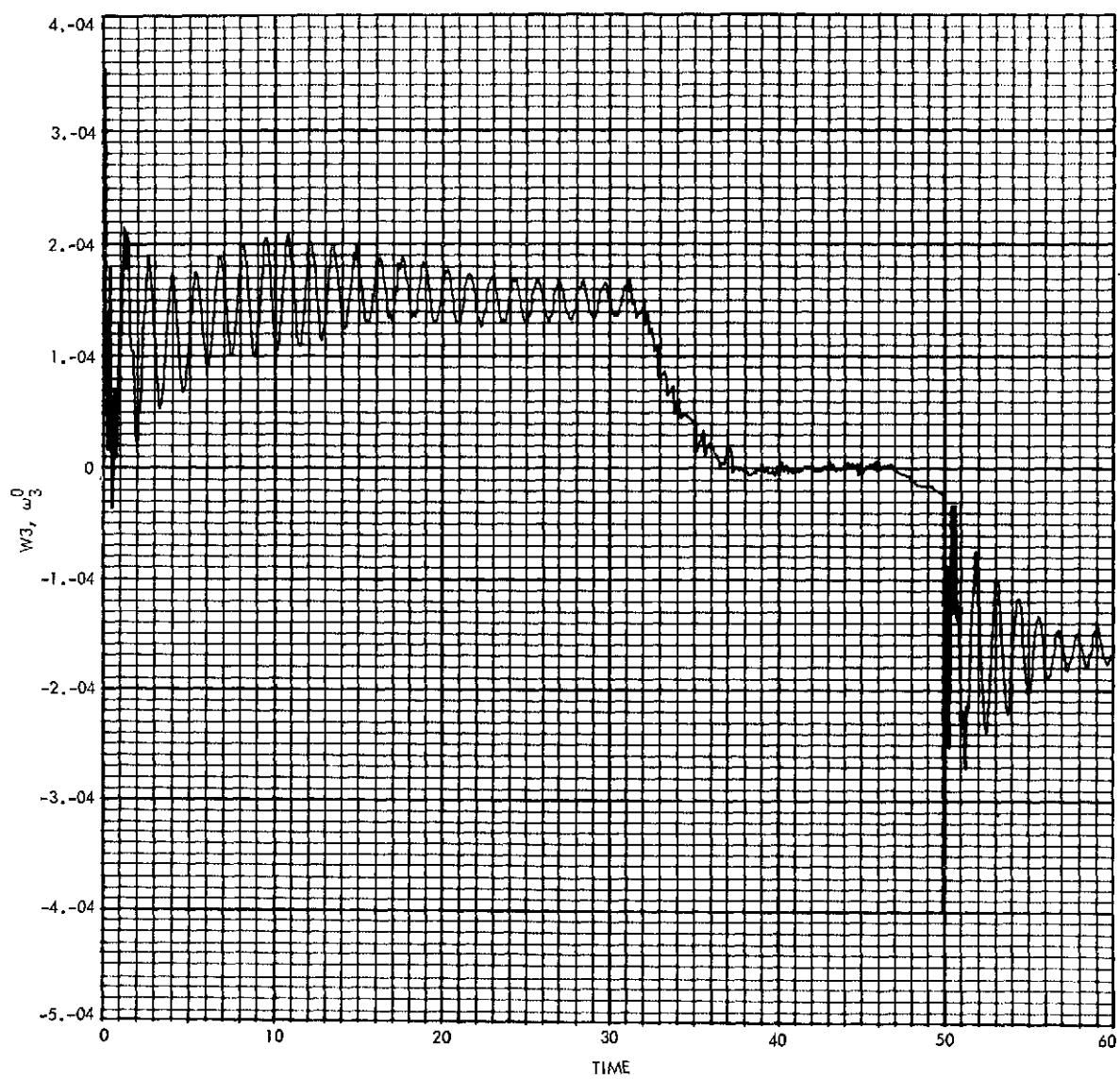


Fig. 7 (contd)

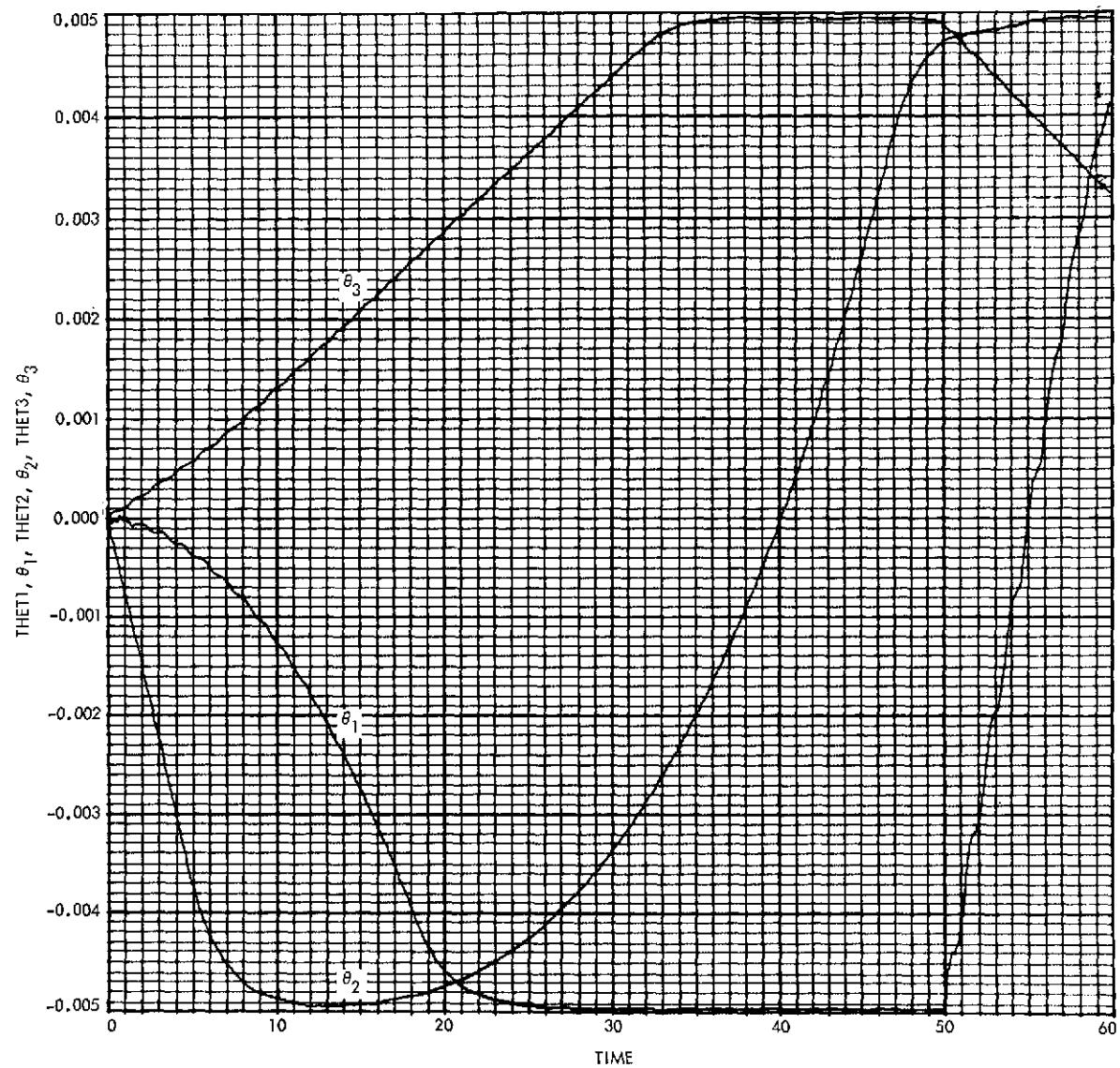


Fig. 7 (contd)

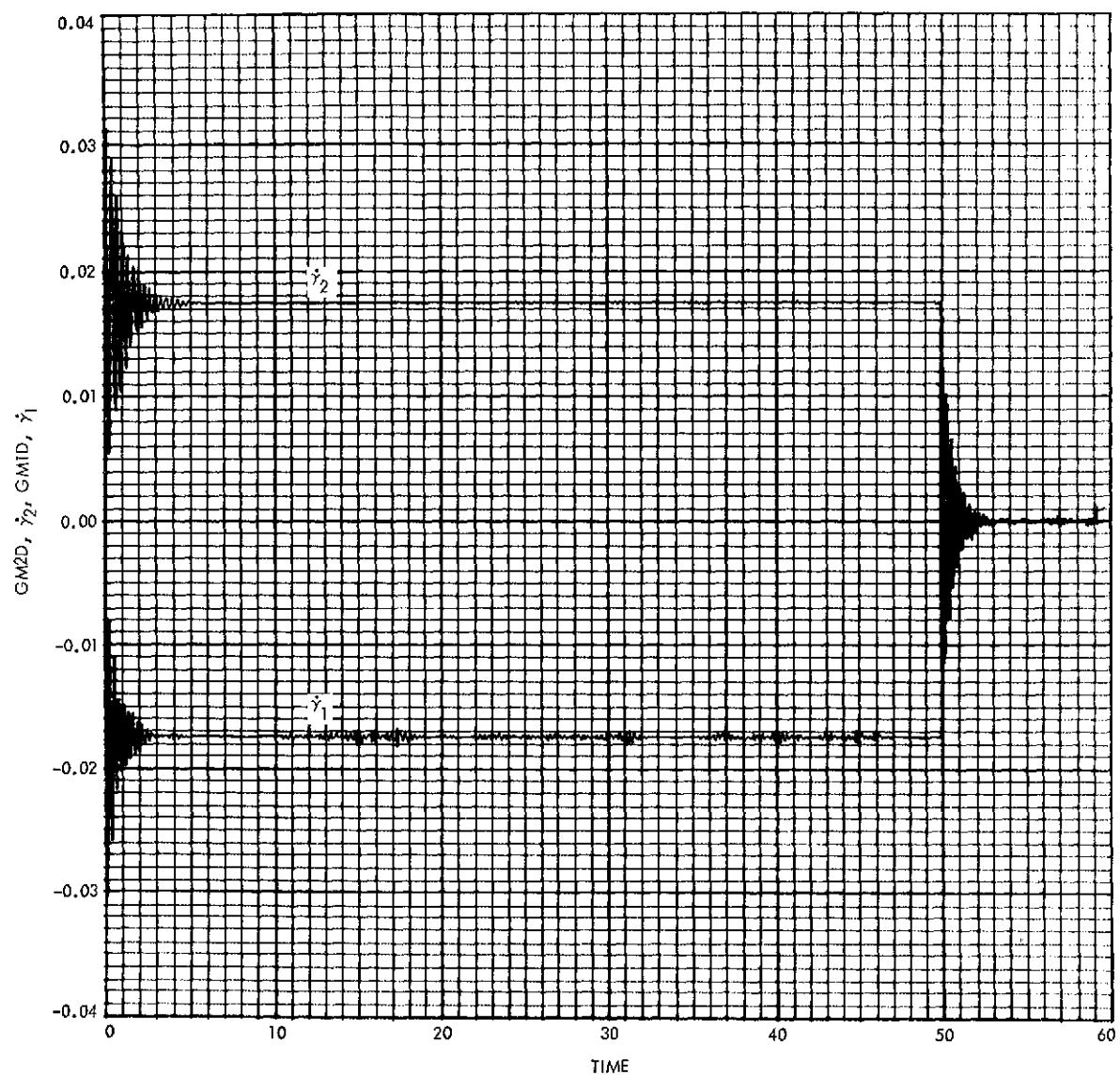


Fig. 7 (contd)

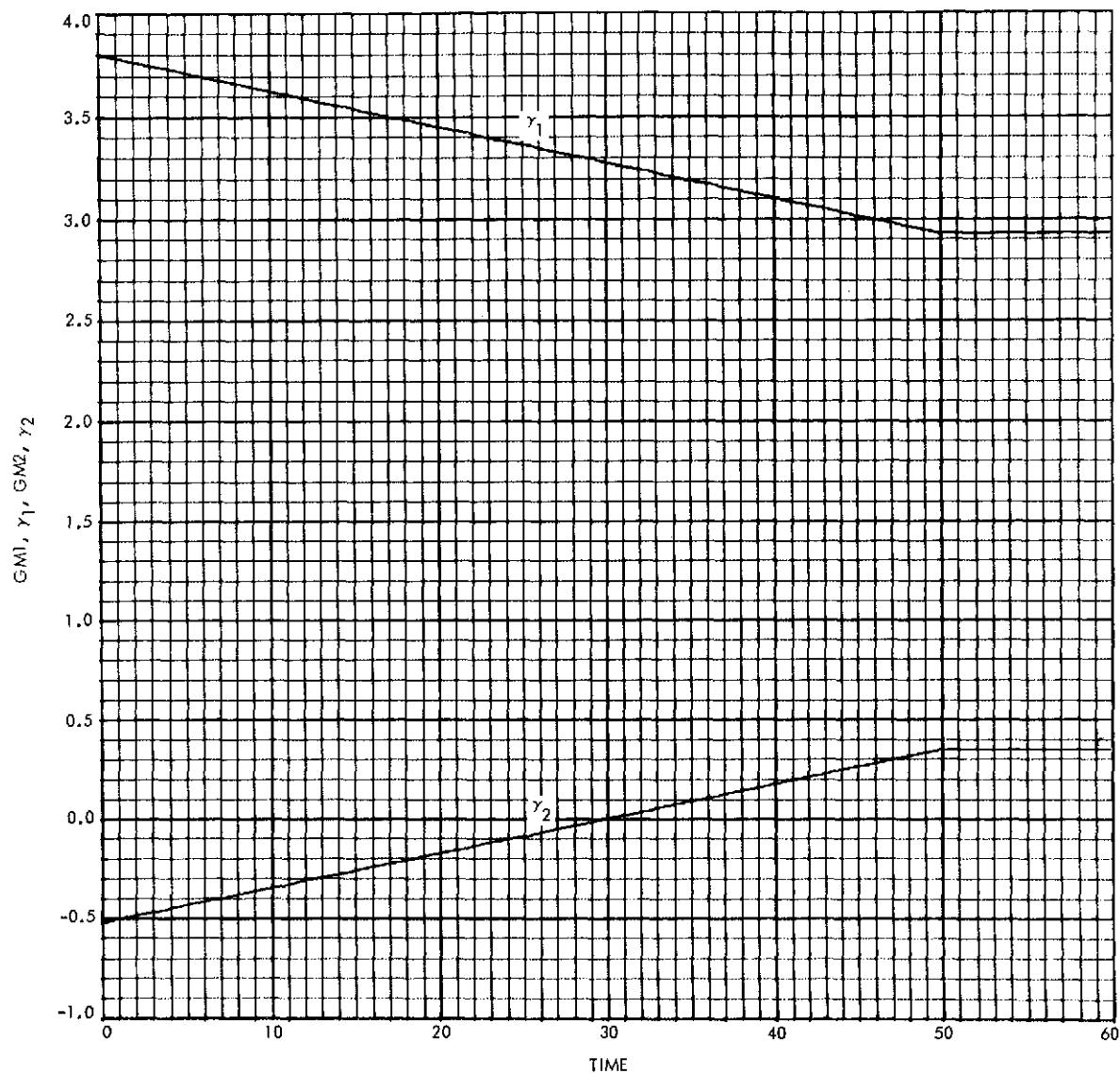


Fig. 7 (contd)

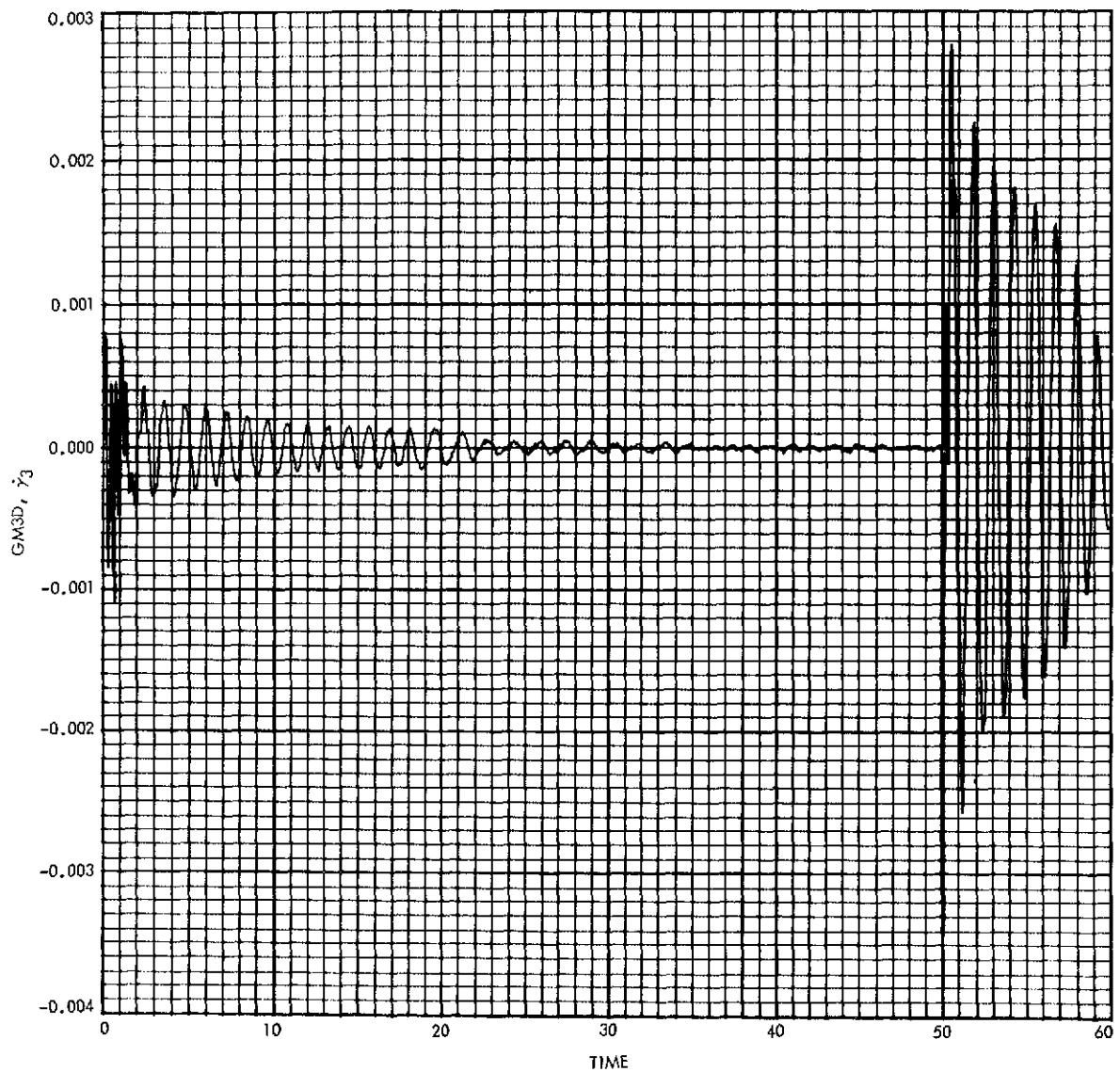


Fig. 7 (contd)

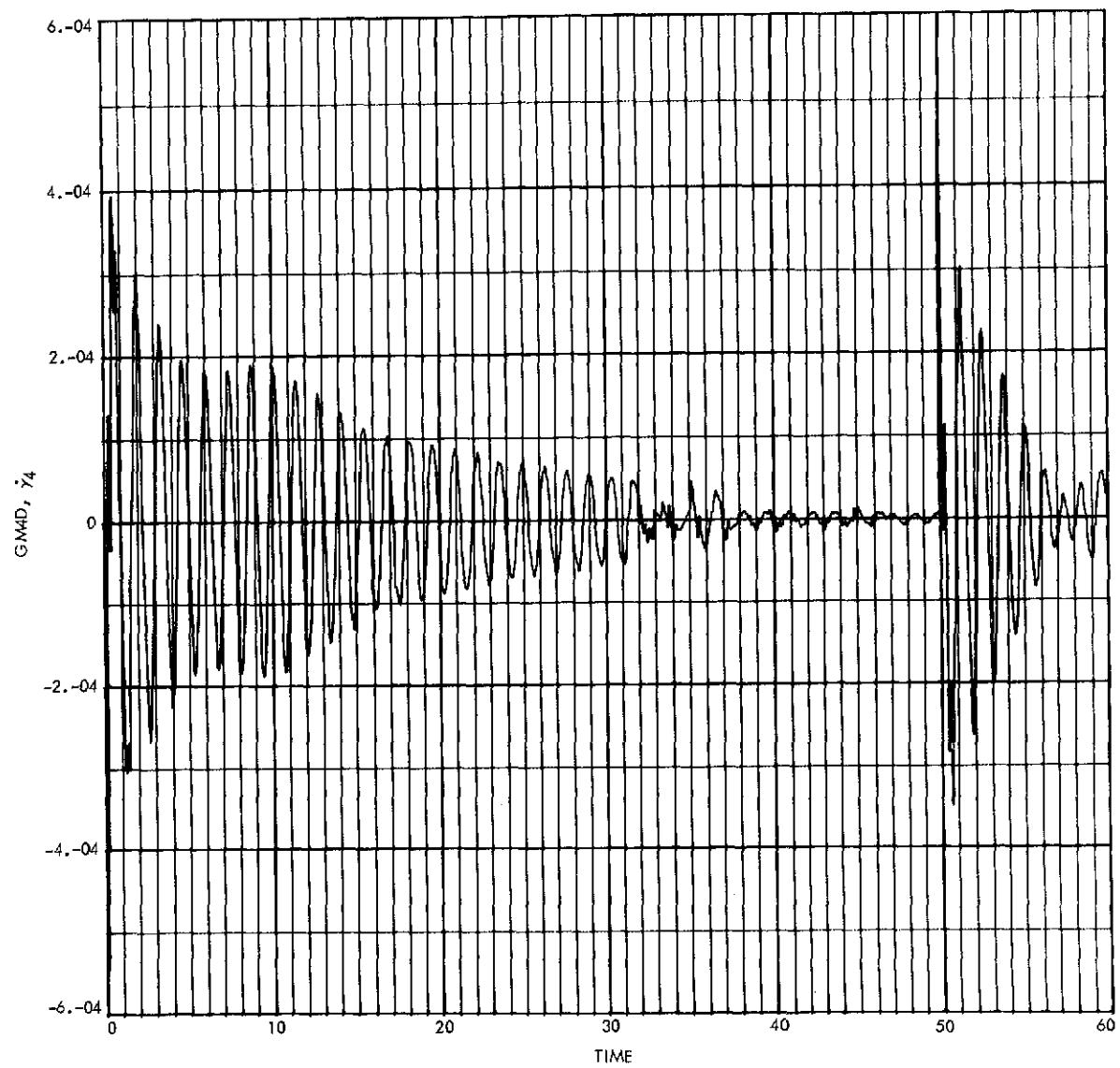


Fig. 7 (contd)

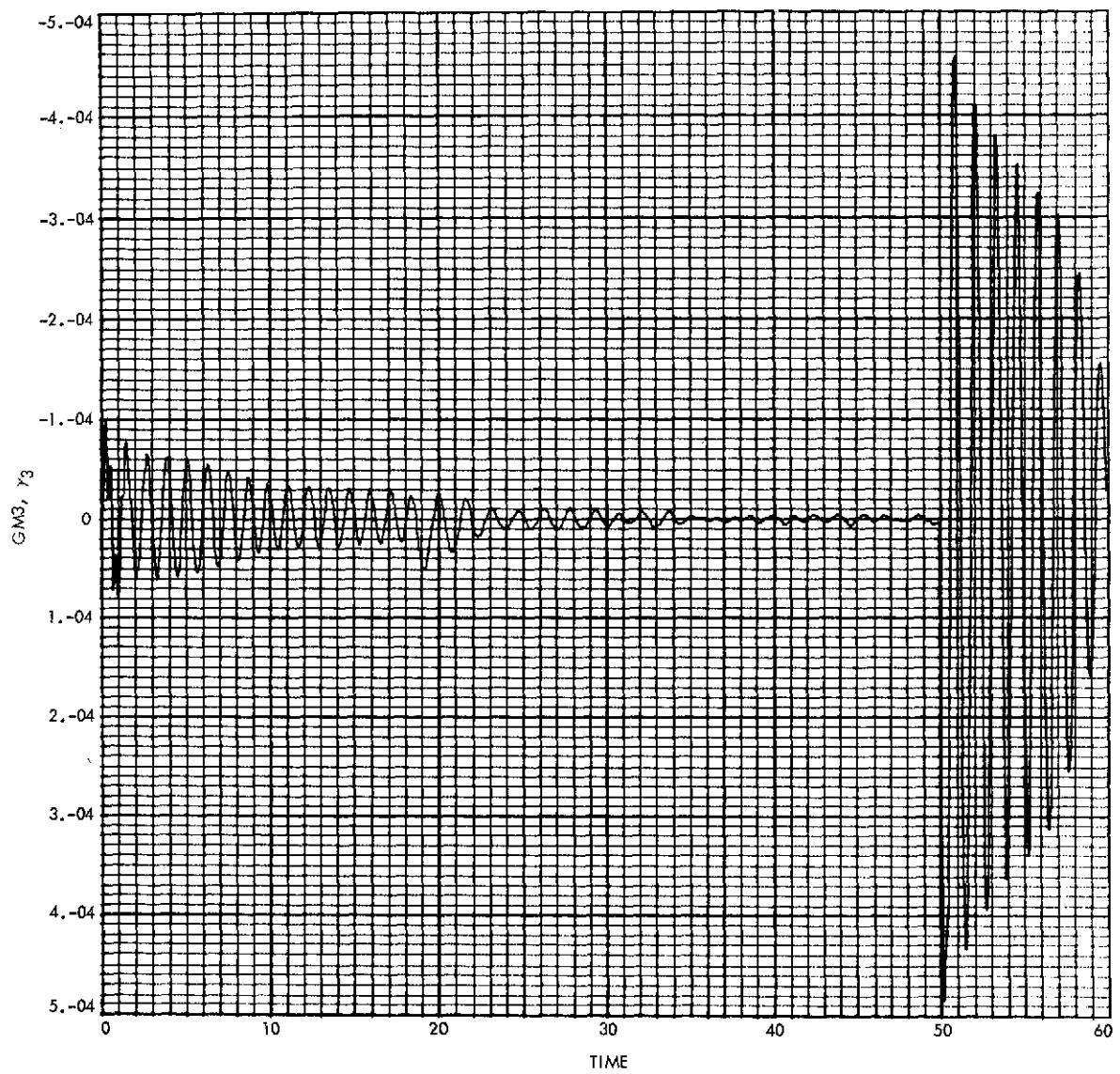


Fig. 7 (contd)

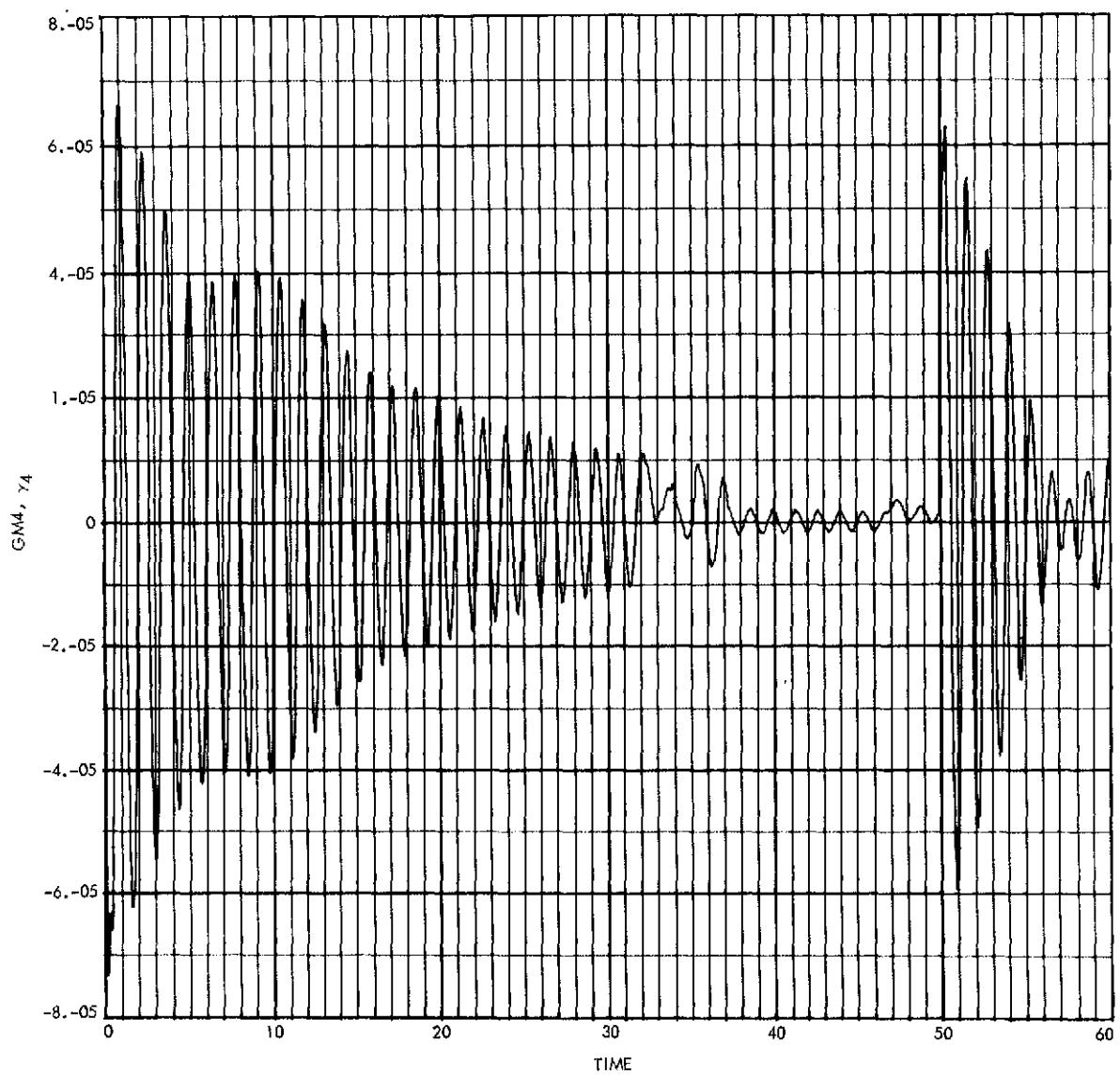


Fig. 7 (contd)

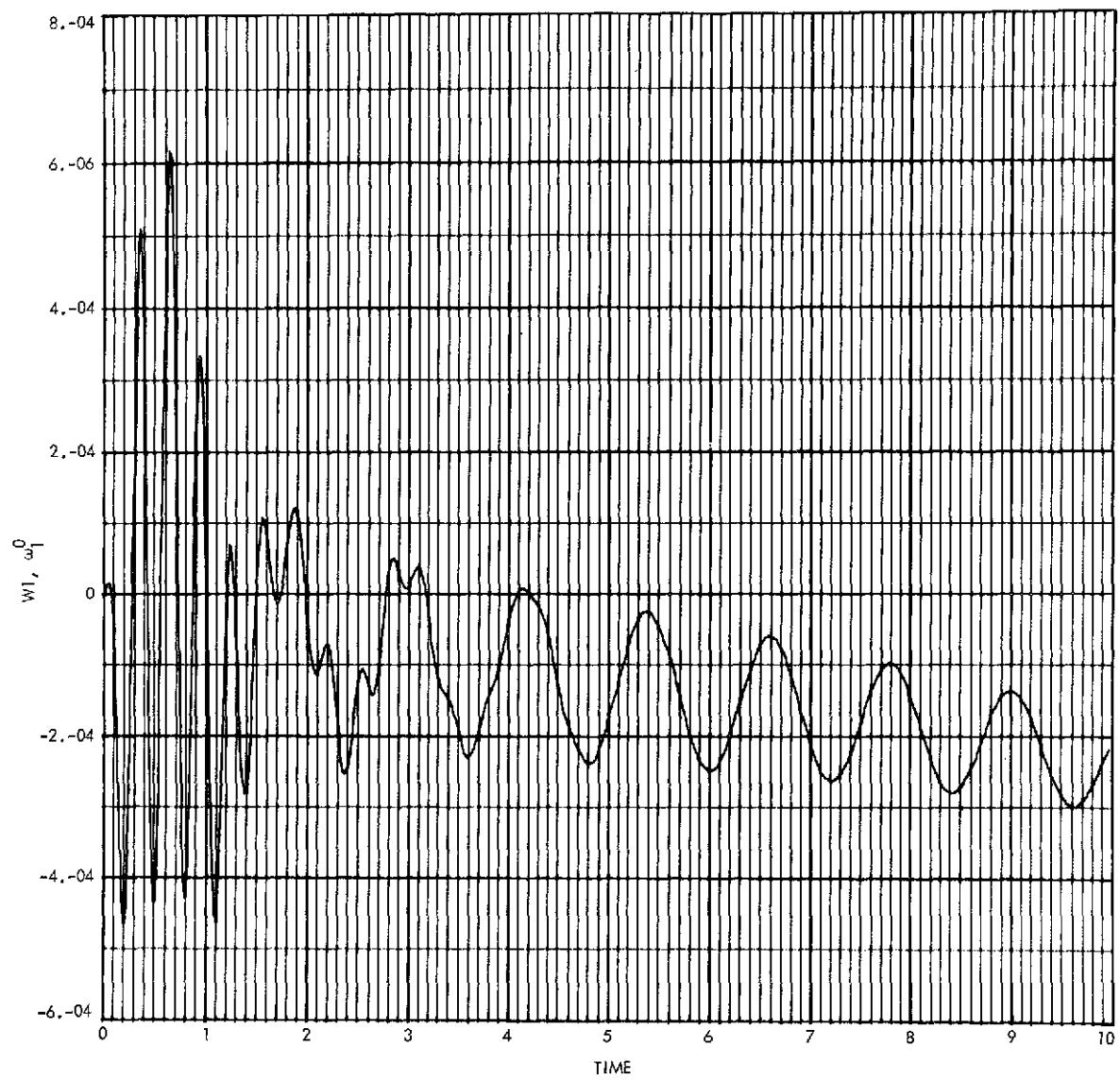


Fig. 8. Results of Case 1 simulation using MBDY (components of ω^0 ; first 10 s)

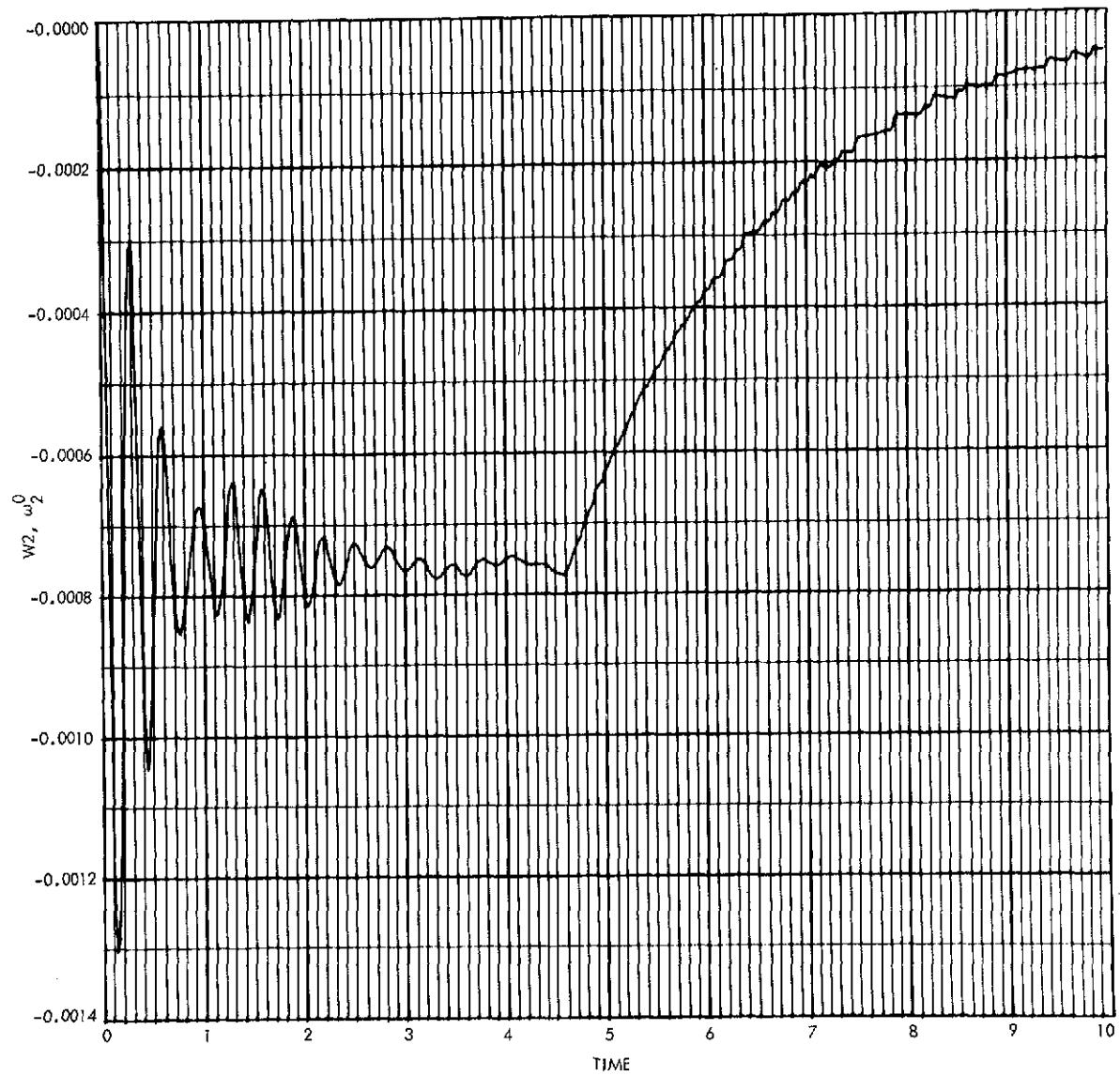


Fig. 8 (contd)

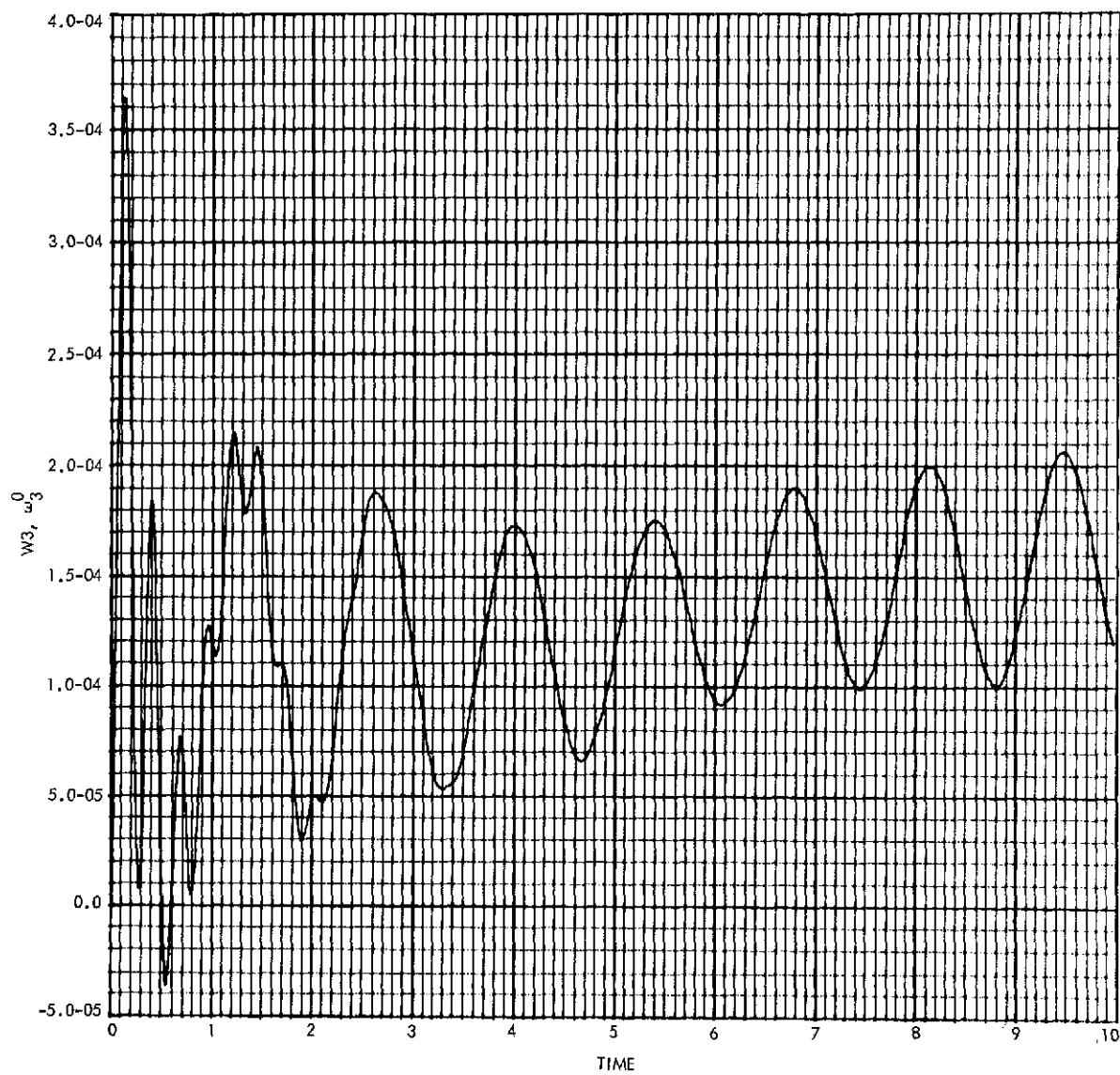


Fig. 8 (contd)

body \mathbf{b}_i , ($i \in \mathcal{P}$) may be prescribed relative to an adjoining body \mathbf{b}_j , where $j = h_i$ (Def. 37), by setting $PI(i) = 1$ and supplying the appropriate values or function for $\dot{\gamma}_i(GMDD(i))$. This was done for the platform bodies \mathbf{b}_1 and \mathbf{b}_2 , resulting in the simulation program of Fig. 9. Notice that the platform command angles have been replaced by time functions for $GMDD(1)$ and $GMDD(2)$ (using the CSSL III "RAMP" function). The $GMDD(1)$ function is plotted in Fig. 10.

The expressions for the hinge torques τ_1 and τ_2 ($TH(1)$, $TH(2)$) have also been removed from the original program (although it was not necessary) to emphasize the fact that they are not used when γ_1 and γ_2 are prescribed.

Figure 11 shows the simulation results. It is clear that the overall system response is virtually the same as that obtained from the first simulation (Fig. 7), except for the absence of the characteristic platform hinge vibrations. Since the simulation with prescribed platform motion took only $4\frac{1}{2}$ min of computation time, compared with $7\frac{1}{2}$ min for the first run, the usefulness of the prescribed motion option is well demonstrated. In this case, not only was the order of the system equation reduced, but the absence of the high-frequency platform vibrations helped to speed up the numerical integration process.

Of course, some caution must also be exercised when employing prescribed variables to make sure that the dynamics being eliminated are not important. In the sample problem considered here, the platform hinge vibrations could well have triggered gas jet firings before damping out if τ_R had been larger and/or the initial conditions had been different. The absence of these vibrations might then have resulted in an entirely different time history of the reference body attitude angles.

One additional simulation of the Case 1 platform maneuver was performed *without* the application of control torques. In the absence of external torques, the system angular velocities should return to their initial values (i.e., zero) after the platform maneuver is completed (Conservation of Angular Momentum). Of particular interest are the reference body angular rates, which are given in Fig. 12. Although vibrations are still in evidence at $t = 60$ s, it is clear that the final *average* rates are zero, as they should be. (The small high-frequency disturbances which seem to occur randomly, particularly in the $W2$ and $W3$ responses, are due to numerical integration errors.)

3. Case 2—small platform maneuver. The subroutine MBDY, since it solves the complete and exact set of equations of motion, must certainly be used when relatively large angles and angular rates are expected at each hinge. However, if the hinge rotations or both the rotations and rotational rates are "small" at any hinge, some linearization of the equations of motion may be effected, often to good advantage in simplifying and speeding the computational task.

In subsequent sections of this report, modified versions of MBDY, employing some form of linearization, will be described. One version, MBDYTL, employs total linearization; i.e., all angular position and velocity responses in the system are assumed "small" in some sense. A sample simulation, using the five-body configuration of Fig. 4, will be performed later using MBDYTL, and it is therefore of interest to have for comparison purposes an "exact" solution to the same problem using MBDY.

```

PROGRAM 5-BODY SYSTEM SIMULATION USING MBDY --- CASE 1
  INTEGER NH,PI(4),H(4)
  ARRAY MB(7),MA(4,7),PB(4,4,3),PA(4,4,3),G(4,3),TB(3),TA(4,3)
  ARRAY FB(3),FA(4,3),TH(4),GM(4),GMD(4),GMDD(4),WD(3)
  DOUBLE PRECISION WDOT(7)
  DATA H/0,1,0,3/PI/1+1,0,0/
  DATA MB/115.,316.,440.,14.,-14.,34.6,410./
  DATA MA(1,1)/.35/MA(1,2)/.35/MA(1,7)/6.8/
  DATA MA(2,1)/4.85/MA(2,2)/2.2/MA(2,3)/5.5/MA(2,4)/-,41/MA(2,5)/.07/
  DATA MA(2,5)/-.54/MA(2,7)/57.5/
  DATA MA(4,1)/27.2/MA(4,2)/.2/MA(4,3)/27.2/MA(4,7)/10.7/
  DATA PB(1,31)/-1.5/PB(3,21)/-1.2/
  DATA PA(1,1,31)/-.75/PA(1,2,21)/-,10/PA(1,2,31)/-.75/
  DATA PA(2,2,21)/-.22/PA(2,2,31)/.20/
  DATA PA(4,4,21)/3.3/
  DATA G(1,31)/1./G(2,1)/-1./G(3,1)/1./G(4,3)/1./
    CONSTANT TM=.020,DB=.005,TR=2.0
    CONSTANT TFINAL=60.,CLKTIM=420.
    CONSTANT L1=.23,L2=.21,L3=.31
    CONSTANT SLRATE=.01745329
    CONSTANT PHIZ=G.,THETZ=0.,PSIZ=0.
    CONSTANT W1I=0.,W2I=0.,W3I=0.
    CONSTANT K1=3500.,K2=3500.,K3=2000.,K4=2000.
    CONSTANT B1=20.,B2=20.,B3=10.,B4=10.
    CONSTANT IC1=218.,IC2=-30.

INITIAL
  NH=4
  RD=57.2957795 $ DR=1./RD $ GM1CI=IC1/RD $ GM2CI=IC2/RD
  GM1I=IC1/RD $ GM2I=IC2/RD
  CALL MBDY(NH,H,MB,MA,PB,PA,G,PI)
  PZI,PAI,PBI,PCI=HCK(INITL,PHIZ,THETZ,PSIZ)
END
DYNAMIC
  IF(TIME.GE.TFINAL) GO TO S1
DERIVATIVE BODY5
  VARIABLE TIME=0.
  CINTERVAL CI=.06
  XERROR W1=1.E-6,W2=1.E-6,W3=1.E-6,GM3=1.E-4
  STPCLK CLK1IM
  OUTPUT 50*W1D,W2D,W3D,W1+W2,W3+THET1,THET2,THET3,,
         IN1,IN2,IN3,GM1,GM2,GM3,GM4,PL1,PL2
  PREPAR W1,W2,W3,THET1,THET2,THET3,GM1D,GM2D,,
         GM1,GM2,GM3D,GM4D,GM3,GM4
NOSORT
  WO(1)=W1 $ WO(2)=W2 $ WO(3)=W3
  PL1=RD*GM1 $ PL2=RD*GM2
  GM(1)=GM1 $ GM(2)=GM2 $ GM(3)=GM3 $ GM(4)=GM4
  GMD(1)=GM1D $ GMD(2)=GM2D $ GMD(3)=GM3D $ GMD(4)=GM4D
COMMENT
COMMENT CELESTIAL-SENSOR-MEASURED ANGLES
COMMENT
  D1*D2=HCK(MATRIX,PZO,PAC,PBO,PCO)
  NX0,NY0,NZ0=HCK(I108,0.,0.,1.,D1,D2)
  LX0,LY0,LZ0=HCK(I108,1.,0.,0.,0.,D1,D2)
  THET1=ATAN2(NY0,NZ0) $ THET2=ATAN2(-LY0,LX0)
  THET3=ATAN2(-LY0,LX0)

```

Fig. 9. Case 1 simulation program using MBDY, with prescribed platform motion

```

COMMENT
COMMENT PSEUDO-RATE FEEDBACK SIGNAL AND SWITCHING AMP. DYNAMICS
COMMENT
    IN1=-THET1-TR*W1 $ IN2=-THET2-TR*W2 $ IN3=-THET3-TR*W3
    AMP1=SWAMP(TIME+TM,DB,IN1)
    AMP2=SWAMP(TIME+TM,DB,IN2)
    AMP3=SWAMP(TIME+TM,DB,IN3)

COMMENT
COMMENT PRESCRIBED PLATFORM ACCELERATIONS
COMMENT
    GMDD(1)=6.981316*(-RAMP10.+TIME)+2.*RAMP1.05,TIME)-...
    RAMP1.1,TIME)+RAMP150.+TIME)-2.*RAMP150.05,TIME)+RAMP150.10,TIME)
    GMDD(2)=GMDD(1)

COMMENT
COMMENT REFERENCE BODY EXTERNAL TORQUE
COMMENT HINGE TORQUES
COMMENT
    TB(1)=L1*AMP1 $ TB(2)=L2*AMP2 $ TB(3)=L3*AMP3
    TH(3)=-K3*GM3-B3*GM3D
    TH(4)=-K4*GM4-B4*GM4D

COMMENT
COMMENT SYSTEM ANGLULAR ACCELERATIONS
COMMENT
    CALL MRATE(NH,TH,TB,TA,FB,FA,GM,GMD,GMDD,W0,W00T)
    W1D=W00T(1) $ W2D=W00T(2) $ W3D=W00T(3)

COMMENT
COMMENT SYSTEM ANGULAR RATES AND POSITIONS
COMMENT
    W1=INTEG(W00T(1),W1I)
    W2=INTEG(W00T(2),W2I)
    W3=INTEG(W00T(3),W3I)
    GM1D=INTEG(W00T(4),0.) $ GM2D=INTEG(W00T(5),0.)
    GM3D=INTEG(W00T(6),0.) $ GM4D=INTEG(W00T(7),0.)
    GM1=INTEG(GM1D,GM1II) $ GM2=INTEG(GM2D,GM2II)
    GM3=INTEG(GM3D,0.) $ GM4=INTEG(GM4D,0.)

COMMENT
COMMENT HCK PARAMETER RATES AND POSITIONS
COMMENT
    PZOD,PAOD,PBOD,PCOD=HCKIHCK,PZ0,PA0,PB0,PC0,W1,W2,W3)
    PZ0=INTEG(PZOD,PZII)
    PA0=INTEG(PAOD,PAI)
    PB0=INTEG(PBOD,PBI)
    PC0=INTEG(PCOD,PCI)

END
END
END
TERMINAL
S1.. CONTINUE.
END
END

```

Fig. 9 (contd)

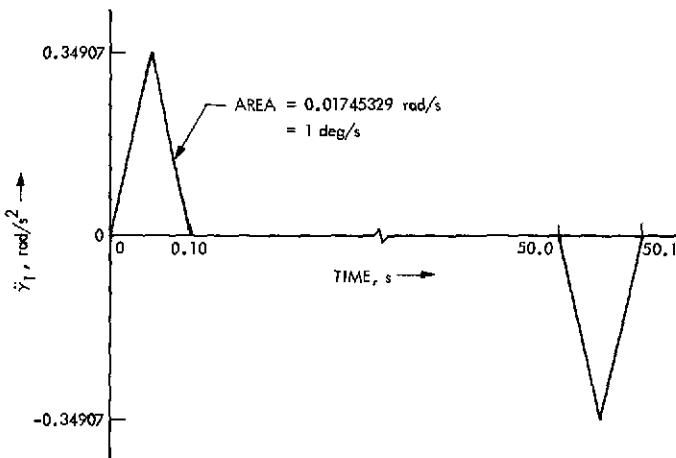


Fig. 10. $\ddot{\gamma}_1$ prescribed function of time ($\ddot{\gamma}_2 = -\ddot{\gamma}_1$)

In Case 1, we have already seen that the reference body and boom responses are relatively small even though the platform angular excursions are substantial. If the platform maneuvers were also restricted to small values, this would seem the ideal case for which to apply MBDYTL and to obtain a check solution using MBDY. Results of the latter will be presented here.

In this case, it will be assumed that the platform again starts from the position $\gamma_1 = 218$ deg and $\gamma_2 = -30$ deg and that at $t = 0$ the command is issued to rotate to $\gamma_1 = 220$ deg at a commanded rate of 1 deg/s. Then, at $t = 5$ s, the command is issued to rotate about the other axis to the position $\gamma_2 = -25$ deg, at the same rate. A CSSL III program to simulate this maneuver sequence is listed in Fig. 13.

The simulation results are shown in Fig. 14. Obviously no gas jets were fired. These solutions will be compared in following sections to those obtained from linearized versions of MBDY.

III. Totally Linearized Systems

A. The Equations

When any justification can be found for the linearization of any of the kinematical variables appearing in Eq. (5), this step should be taken, since the mathematical and computational benefits of linearization are quite substantial. The primary immediate benefit is the elimination of variable terms from the coefficient matrix on the left side of Eq. (5). If ω^0 and all angles $\gamma_1, \dots, \gamma_n$ and their time derivatives can be assumed to remain *arbitrarily* small, then the indicated coefficient matrix is formally constant, and much computer simulation time can be gained by computing its inverse only once, in advance of numerical integration. When the angles are expected to be small in an engineering sense, less rigorous arguments can be marshalled to gain the same computational advantage. However, more than sufficient reason for employing linearization can very often be found by experimental demonstration.

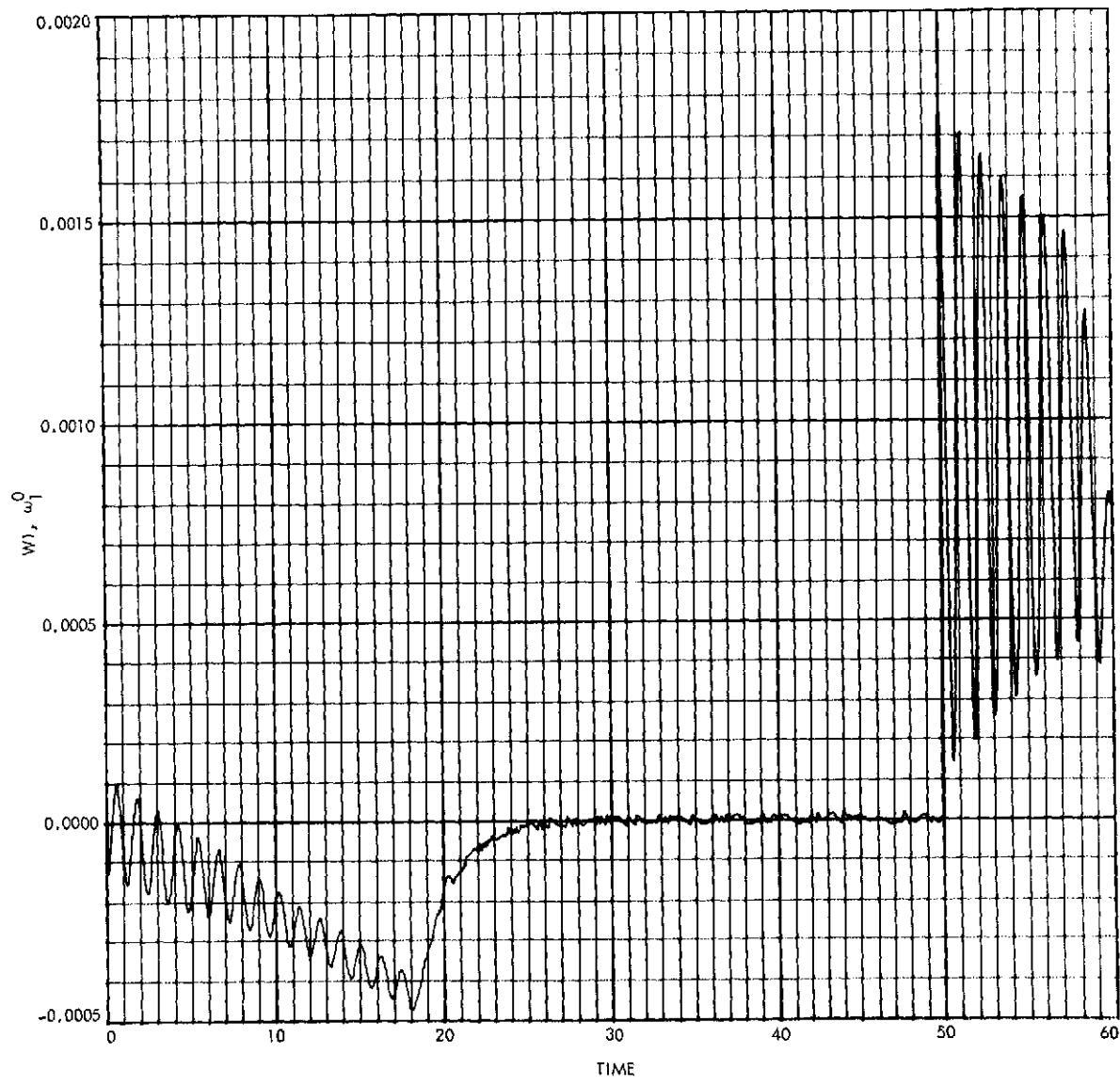


Fig. 11. Results of Case 1 simulation with prescribed platform motion

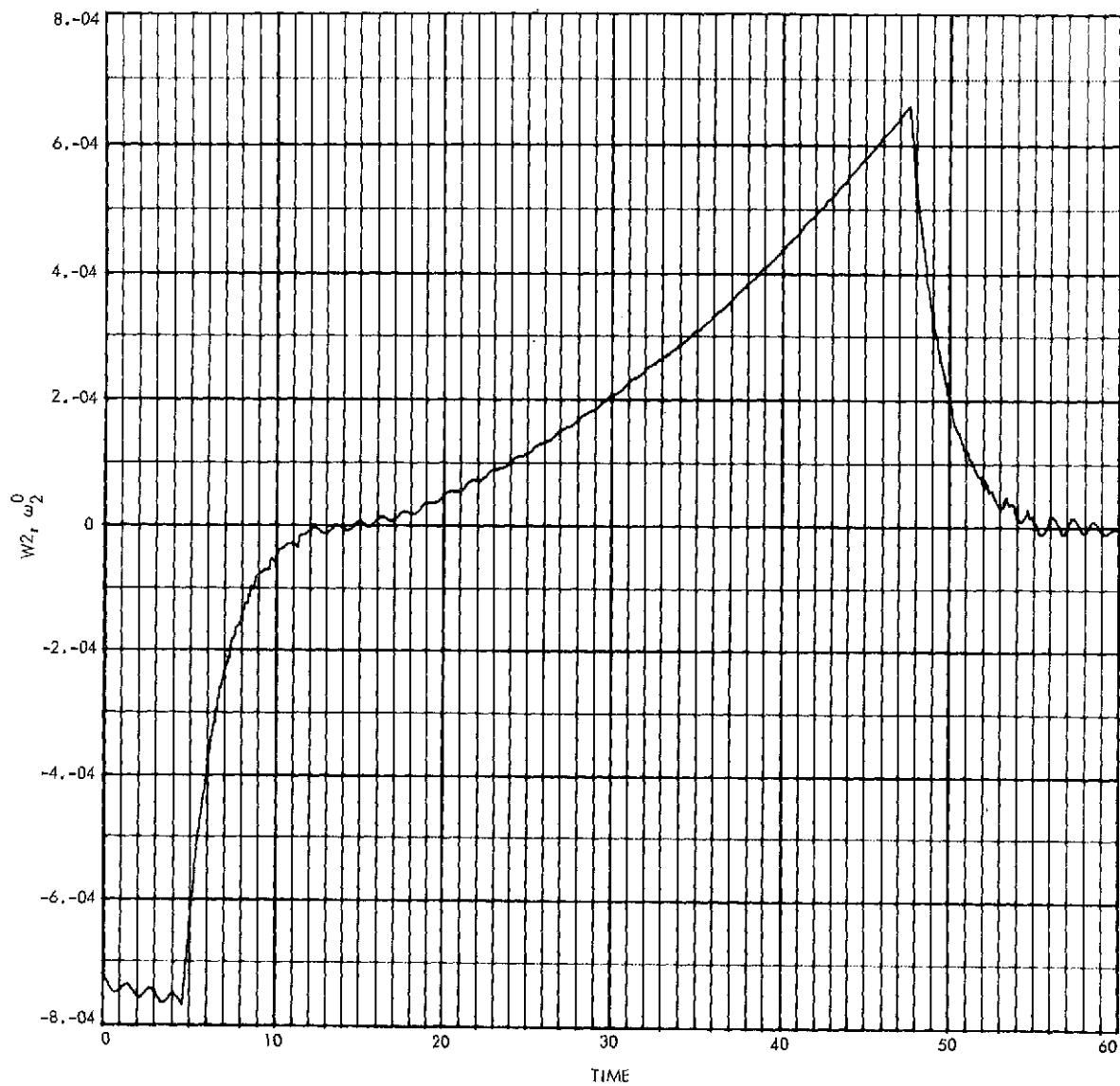


Fig. 11 (contd)

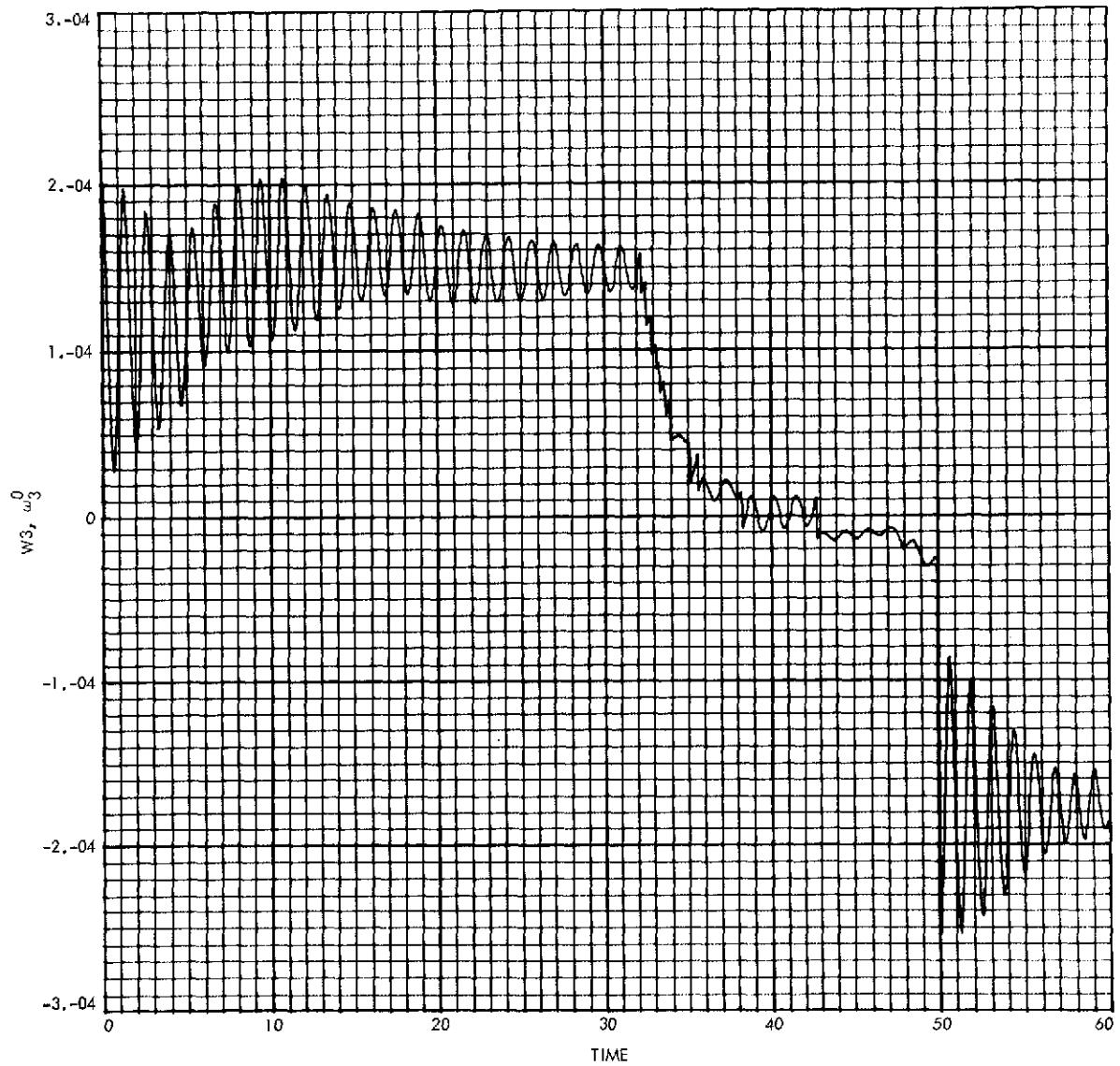


Fig. 11 (contd)

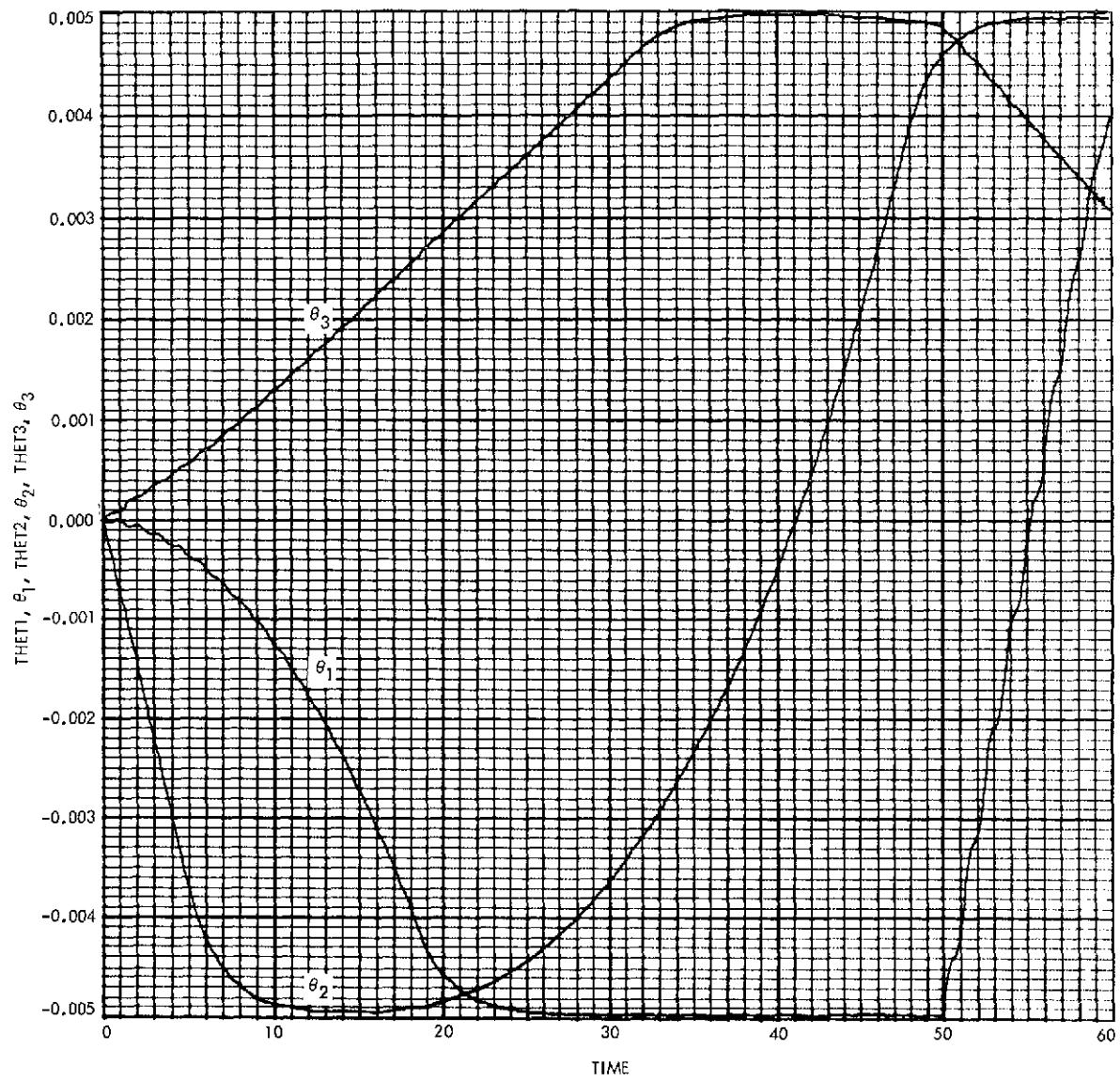


Fig. 11 (contd)

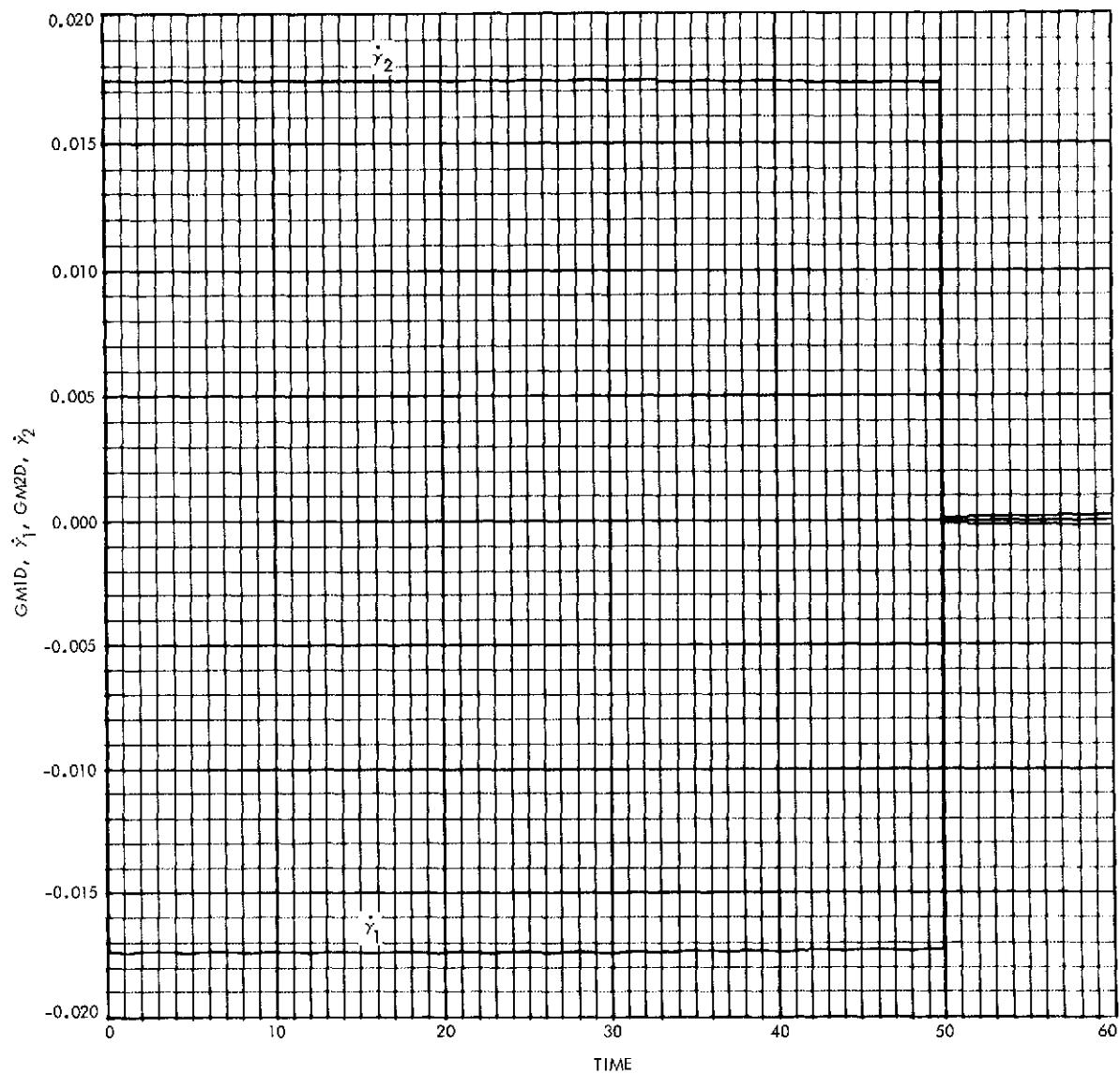


Fig. 11 (contd)

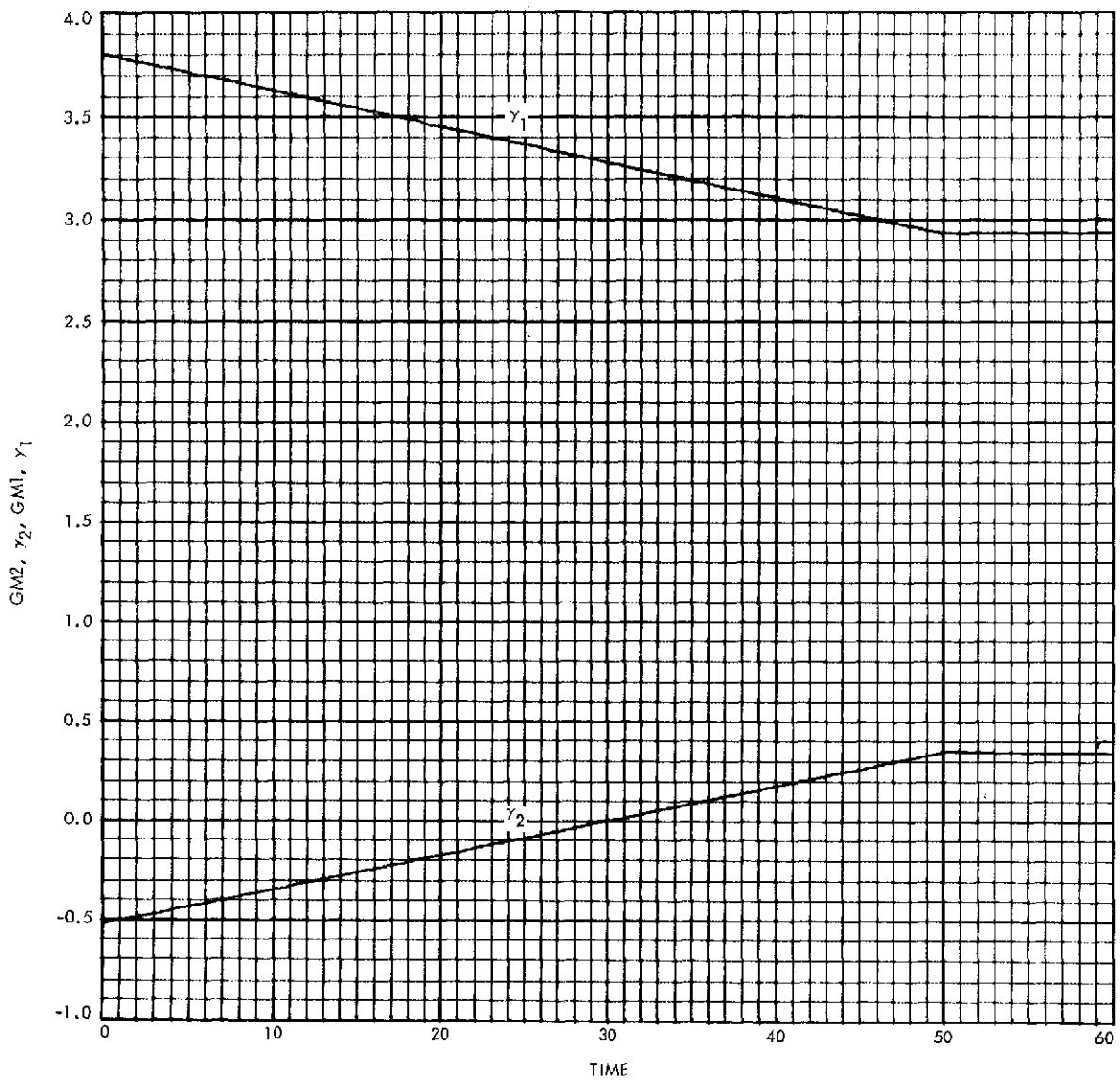


Fig. 11 (contd)

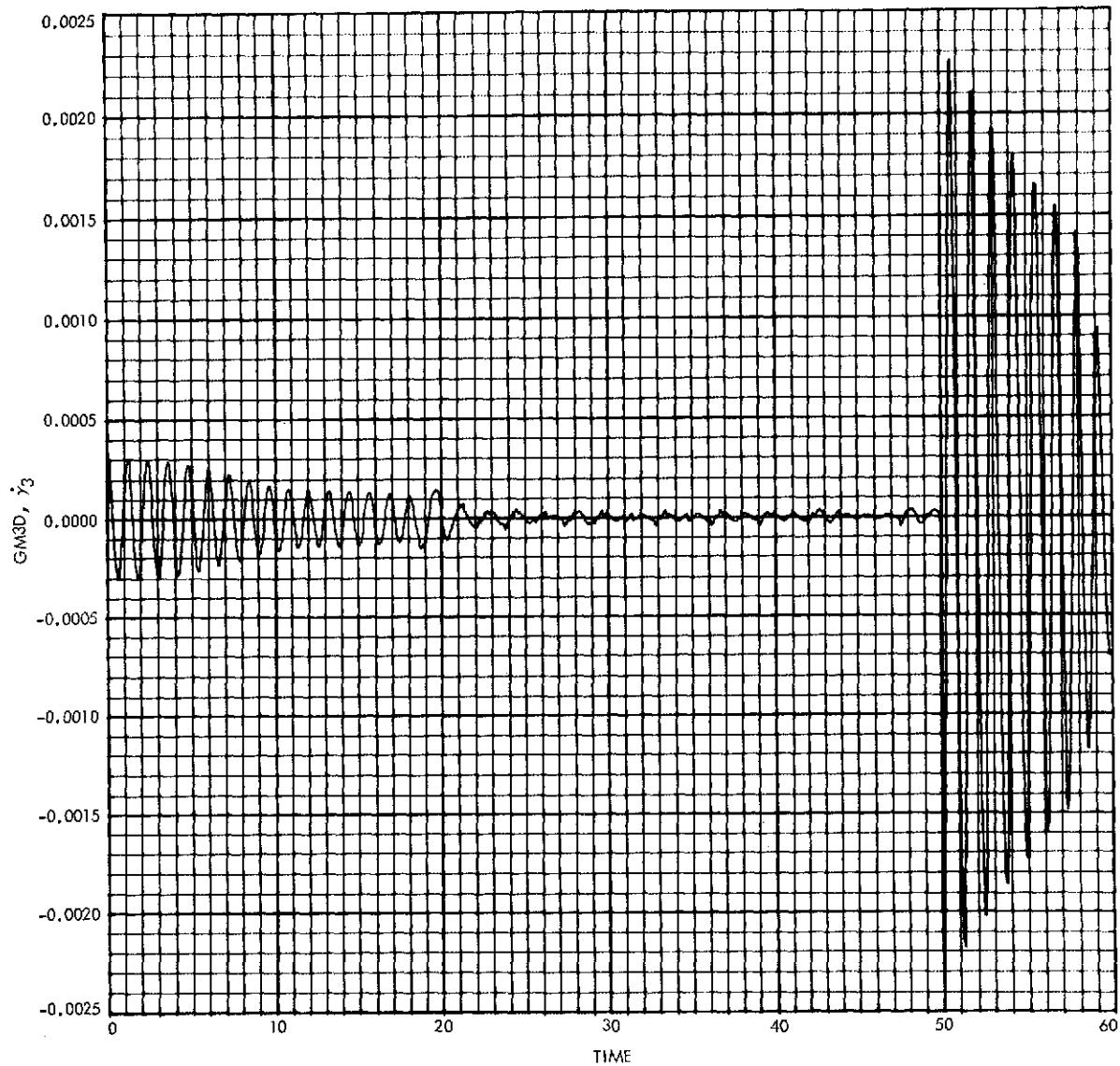


Fig. 11 (contd)

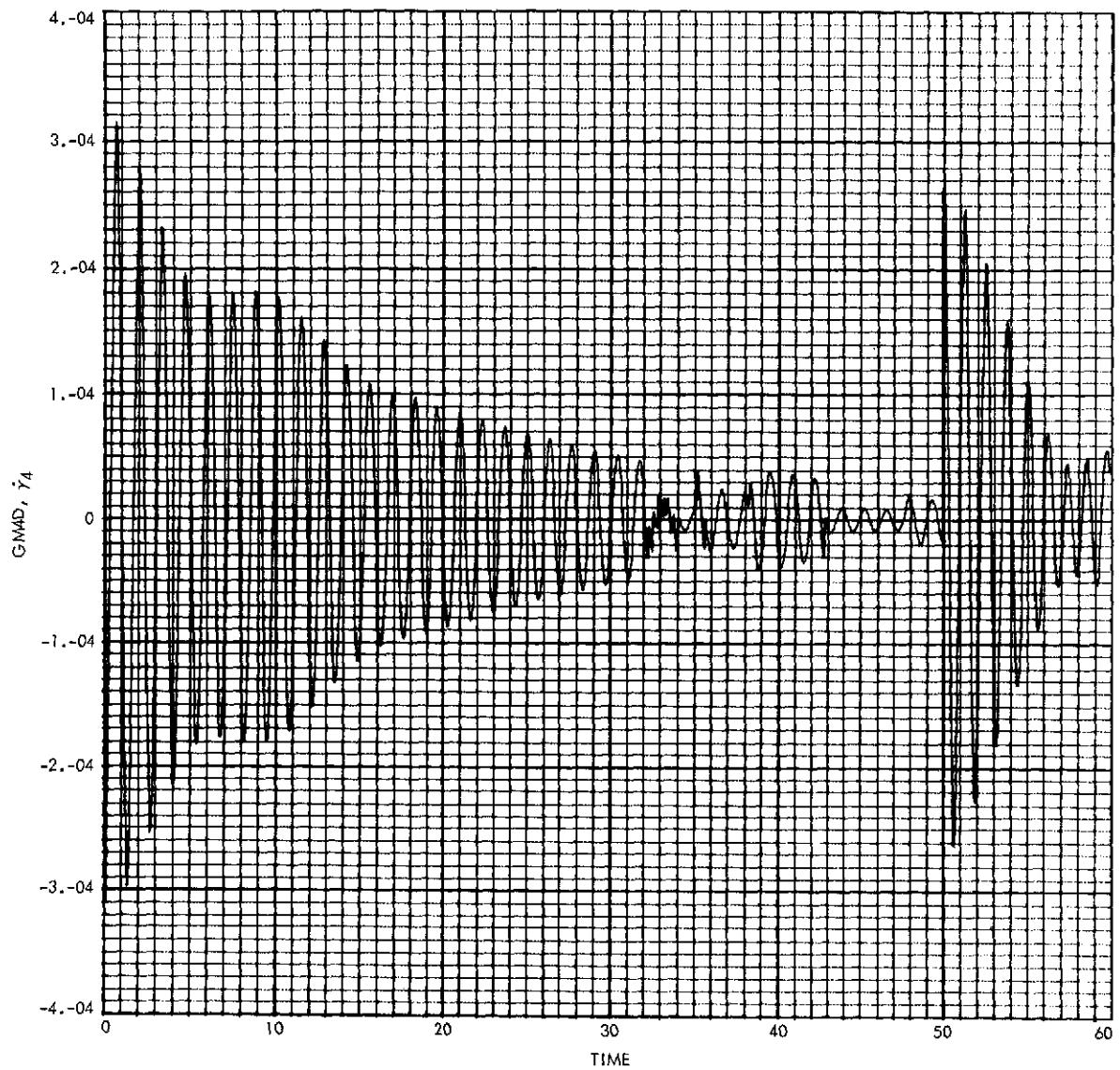


Fig. 11 (contd)

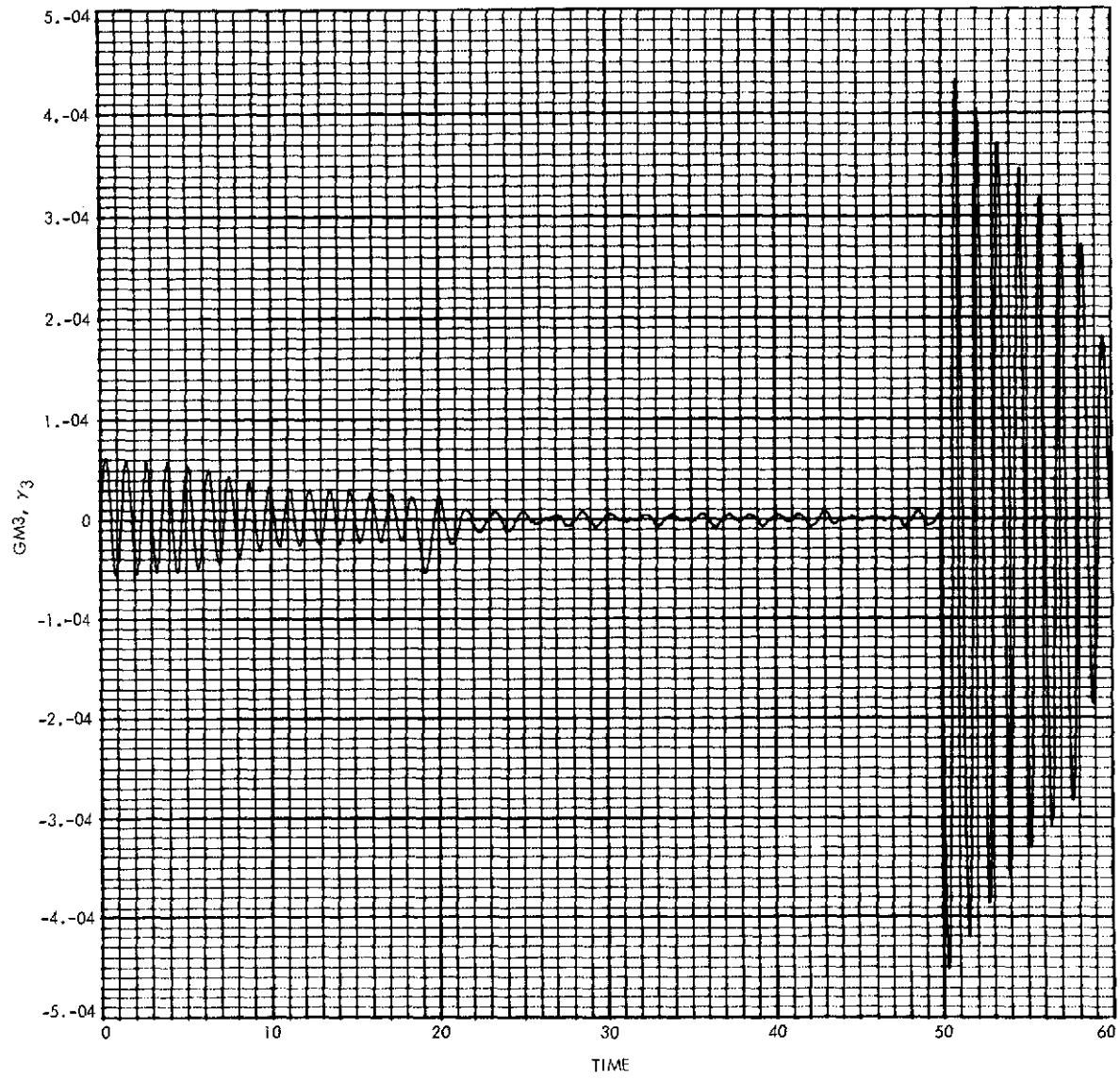


Fig. 11 (contd)

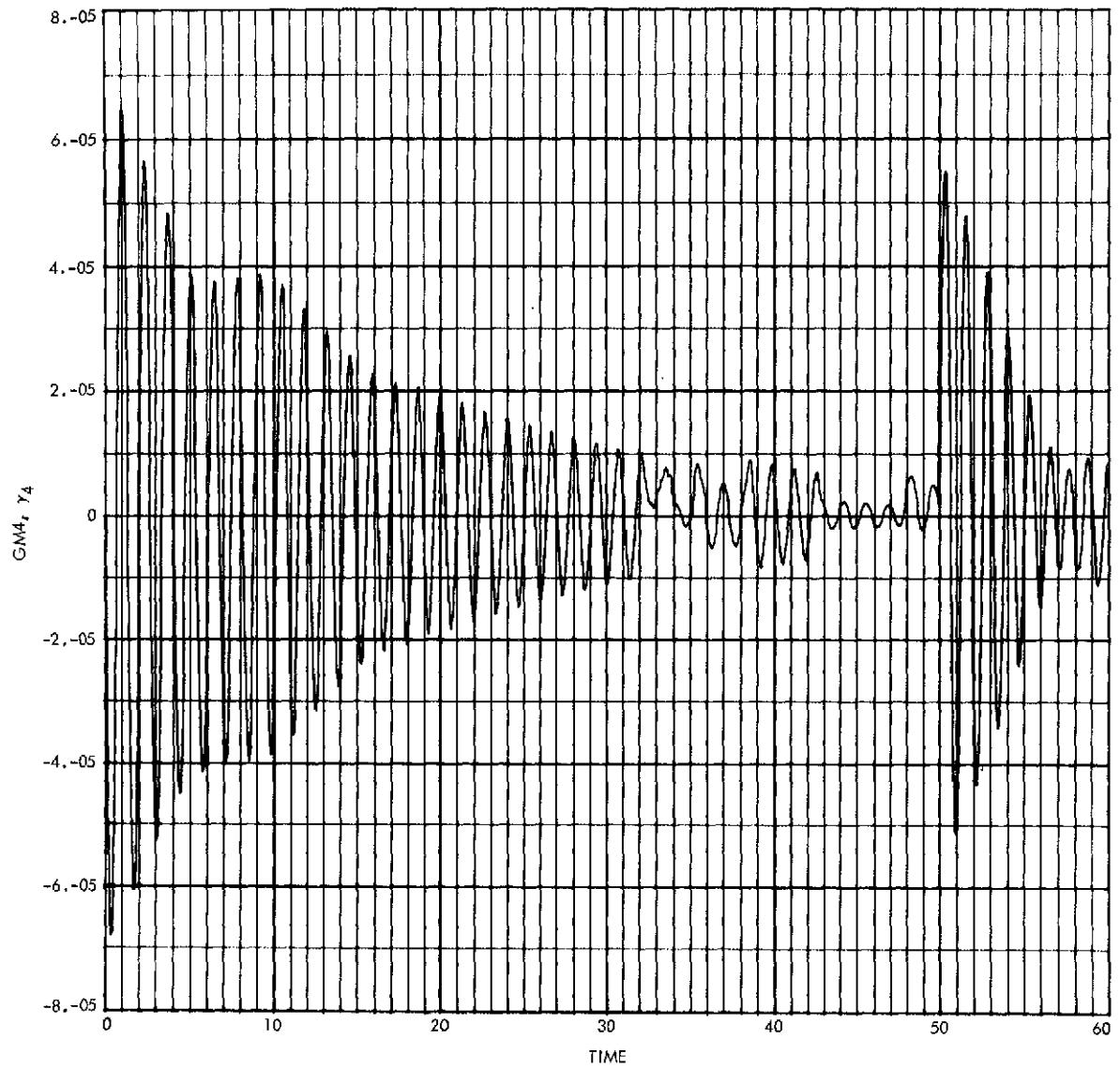


Fig. 11 (contd)

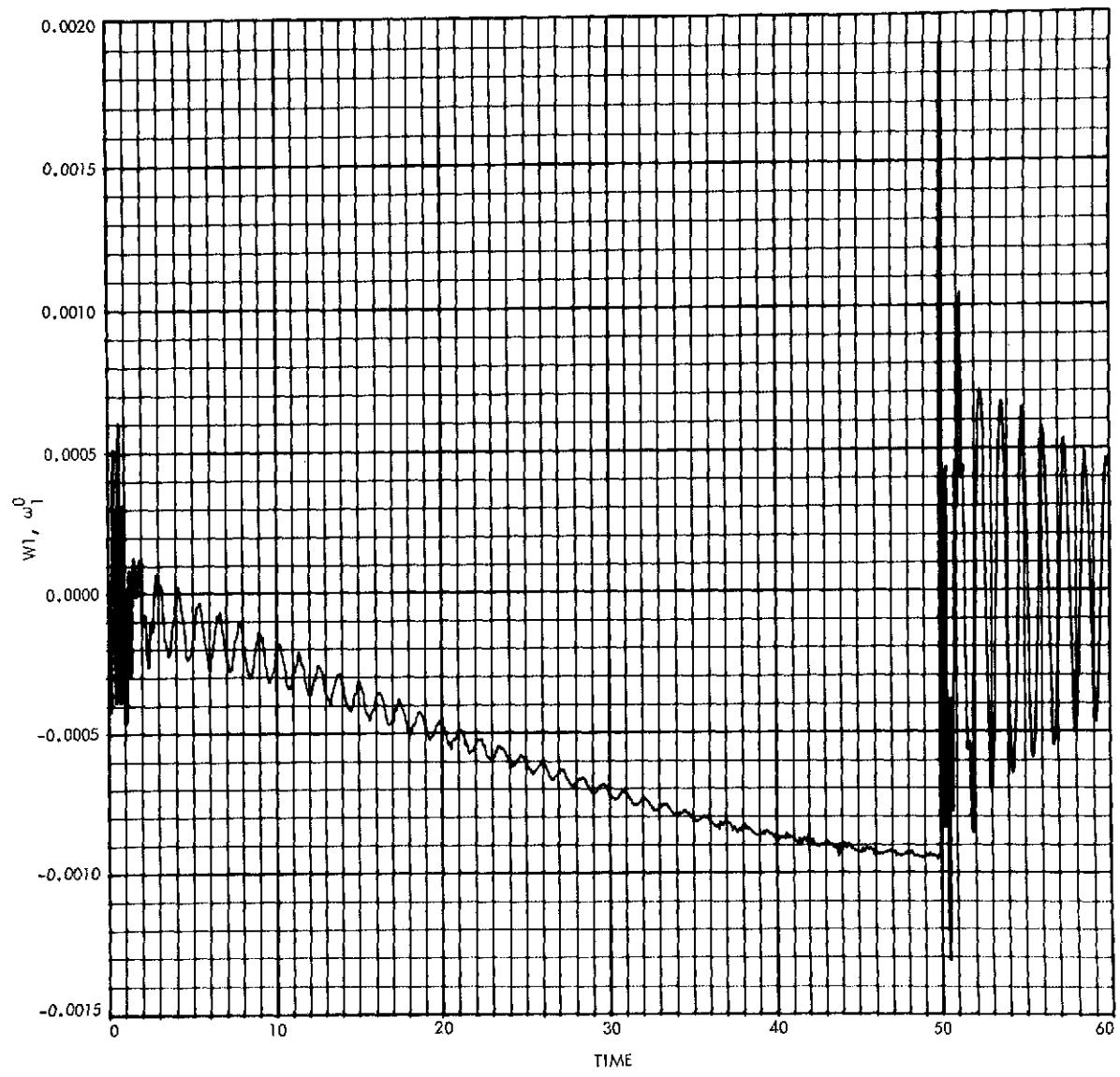


Fig. 12. Results of Case 1 simulation without control torques

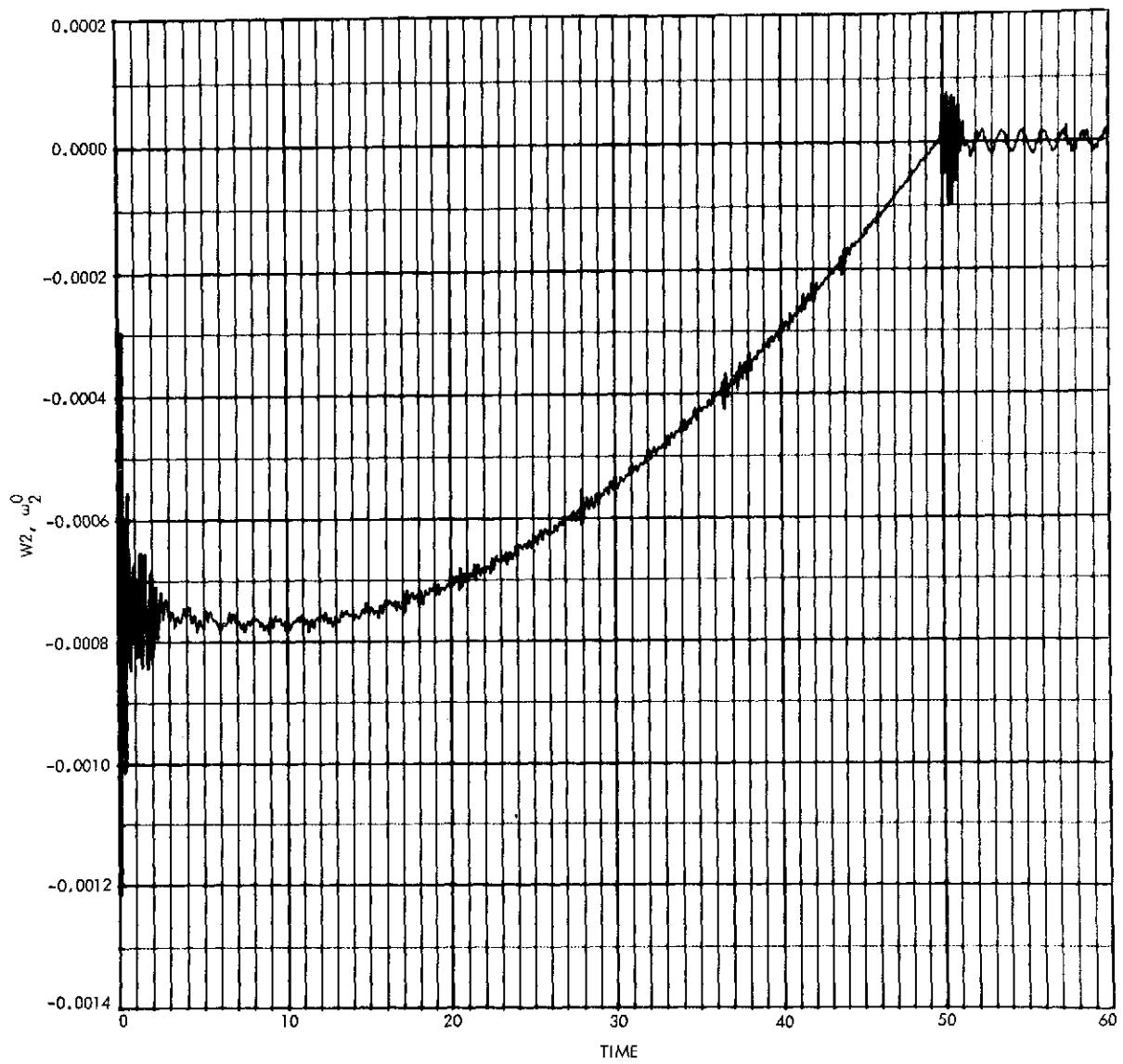


Fig. 12 (contd)

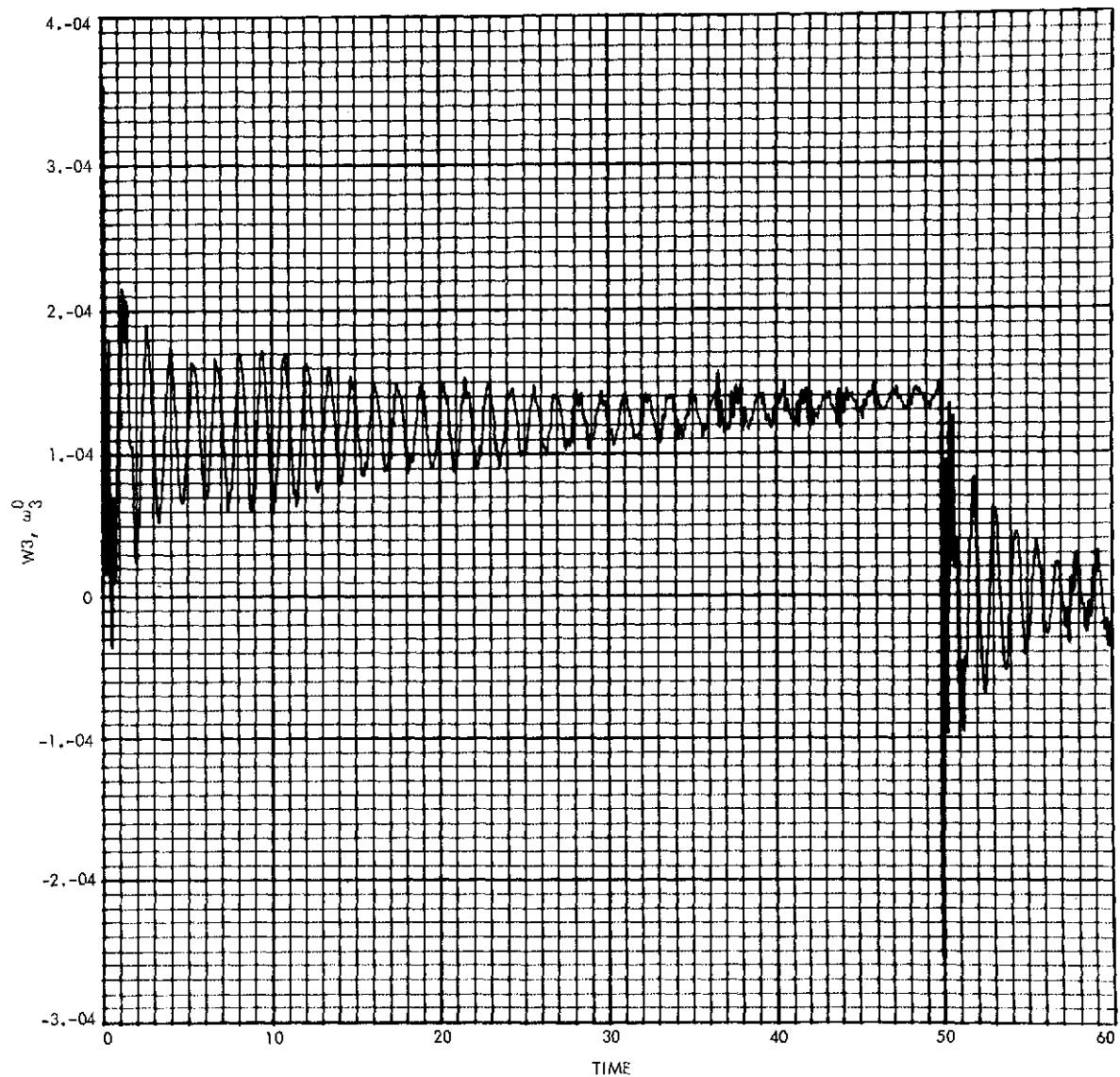


Fig. 12 (contd)

```

PROGRAM S-BODY SYSTEM SIMULATION USING MBDY --- CASE 2
  INTEGER NH,PI(4),H(4)
  ARRAY MB(3),MA(4,7),PB(4,3),PA(4,4,3),G(4,3),TB(3),TA(4,3)
  ARRAY FB(3),FA(4,3),TH(4),GM(4),GMD(4),W(4,3)
  DOUBLE PRECISION WDCT(7)
  DATA H/0.1,0.3/PI/0,0,0/
  DATA MB/115.,316.,440.,14.,-14.,34.,6,410./
  DATA MA1,1/-.35/MA1,2/,.35/MA1,7/6.8/
  DATA MA12,1/4.85/MA12,2/2.2/MA12,3/5.5/MA12,4/-.41/MA12,5/1.07/
  DATA MA12,6/-.54/MA12,7/51.5/
  DATA MA14,1/27.2/MA14,2/.2/MA14,3/27.2/MA14,7/10.7/
  DATA PB1,3/-1.5/PB1,2/-1.2/
  DATA PA1,1,3/-.75/PA1,2,2/-1.10/PA1,2,3/-1.75/
  DATA PA12,2,2/-.22/PA12,2,3/-.20/
  DATA PA14,4,2/3.3/
  DATA G(1,3)/1./G(1,2,1)/-1./G(3,1)/1./G(4,3)/1./
  CONSTANT TM=.020,DB=.005,TR=2.0
  CONSTANT TFINAL=20.,CLKTIM=1100.
  CONSTANT L1=-.23,L2=-.21,L3=-.33
  CONSTANT SLRATE=.01745329
  CONSTANT PHIZ=0.,THETZ=0.,PSIZ=0.
  CONSTANT W1I=0.,W2I=0.,W3I=0.
  CONSTANT K1=3500.,K2=3500.,K3=2000.,K4=2000.
  CONSTANT B1=20.,B2=20.,B3=10.,B4=10.
  CONSTANT IC1=218.,IC2=-30.

INITIAL
  NH=4
  RD=57.2957795 $ DR=1./RD $ GM1C1=IC1/RD $ GM2C1=IC2/RD
  GM1I=IC1/RD $ EM2I=IC2/RD
  CALL MBDV(NH,H,MB,MA,PB,PA,G,PI)
  PZI:PAI:PB1:PC1=HCK(INITL,PHIZ,THETZ,PSIZ)

END
DYNAMIC
  IF(TIME.GE.TFINAL) GO TO S1
DERIVATIVE BODY5
  VARIABLE TIME=0.
  CINTERVAL CI=.02
  XERROR W1=1.E-6,W2=1.E-6,W3=1.E-6,GM3=1.E-9
  STPCLK CLKTFP
  OUTPUT 50,W1D,W2D,W3D,W1,W2,W3,THET1,THET2,THET3,....
  IN1,IN2,IN3,GM1,GM2,GM3,GM4,PL1,PL2
  PREPAR W1,W2,W3,THET1,THET2,THET3,GM1D,GM2D,....
  GM1,GM2,GM3D,GM4D,GM3,GM4
NOSORT
  W0(1)=W1 $ W0(2)=W2 $ W0(3)=W3
  PL1=RD*GM1 $ PL2=RD*GM2
  GM1I=GM1 $ EM2I=GM2 $ GM13=GM3 $ GM14=GM4
  GMD11=GM1D $ GMD21=GM2D $ GMD13=GM3D $ GMD14=GM4D
COMMENT
COMMENT CELESTIAL-SENSOR-MEASURED ANGLES
COMMENT
  D1,D2=HCK(MATRIX,PAC,PBD,PCD)
  NX0,NY0,NZ0=HCK(IITOB,0.,0.,1.,01,02)
  LX0,LY0,LZ0=HCK(IITOB,1.,0.,0.,01,02)
  THET1=ATAN2(NY0,NZ0) $ THET2=ATAN2(-NX0,NZ0)
  THET3=ATAN2(-L10,LX0)

```

Fig. 13. Case 2 simulation program using MBDY

```

COMMENT
COMMENT      RATE FEEDBACK SIGNAL AND SWITCHING AMP. DYNAMICS
COMMENT
IN1=-THET1-TR*#1   $ IN2=-THET2-TR*W2   $ IN3=-THET3-TR*W3
AMP1=SWAMP{TIME+TM,DB,IN1}
AMP2=SWAMP{TIME+TM,DB,IN2}
AMP3=SWAMP{TIME+TM,DB,IN3}

COMMENT
COMMENT      PLATFORM POSITION COMMANDS
COMMENT
GM1CD=SLRAT*(1+-STEP12+,TIME)  $ GM1C=INTEG(GM1CD,GM1CI)
GM2CD=SLRAT*(STEP15+,TIME)-STEP10+,TIME) $ GM2C=INTEG(GM2CD,GM2CI)

COMMENT
COMMENT      REFERENCE EDDY EXTERNAL TORQUE
COMMENT      HINGE TORQUES
COMMENT
TB1(1)=L1*AMP1   $ TB1(2)=L2*AMP2   $ TB1(3)=L3*AMP3
TH1(1)=-K1*(GM1C)-B1*GM1D
TH1(2)=-K2*(GM2C)-B2*GM2D
TH1(3)=-K3*(GM3C)-B3*GM3D
TH1(4)=-K4*(GM4C)-B4*GM4D

COMMENT
COMMENT      SYSTEM ANGULAR ACCELERATIONS
COMMENT
CALL MRATE(NH,TH,TB,TA,FB,FA,GM,GMD,GMD0,W0,WDOT)
W1D=WDOT(1)   $ W2D=WDOT(2)   $ W3D=WDOT(3)

COMMENT
COMMENT      SYSTEM ANGULAR RATES AND POSITIONS
COMMENT
W1=INTEG(WDOT(1),W1I)
W2=INTEG(WDOT(2),W2I)
W3=INTEG(WDOT(3),W3I)
GM1D=INTEG(WDOT(4),0.)    $ GM2D=INTEG(WDOT(5),0.)
GM3D=INTEG(WDOT(6),0.)    $ GM4D=INTEG(WDOT(7),0.)
GM1=INTEG(GM1D,GM1II)    $ GM2=INTEG(GM2D,GM2II)
GM3=INTEG(GM3D,0.)        $ GM4=INTEG(GM4D,0.)

COMMENT
COMMENT      HCK PARAMETER RATES AND POSITIONS
COMMENT
PZCD,PAOD,PBOD,PCOD=HCK(PCK,PZO,PAO,PBO,PCO,W1,W2,W3)
PZO=INTEG(PZOD,PZI)
PAO=INTEG(PAOD,PAI)
PBOD=INTEG(PBOD,PBI)
PCOD=INTEG(PCOD,PCI)

END
END
END
TERMINAL
S1..  CONTINUE
END
END

```

Fig. 13 (contd)

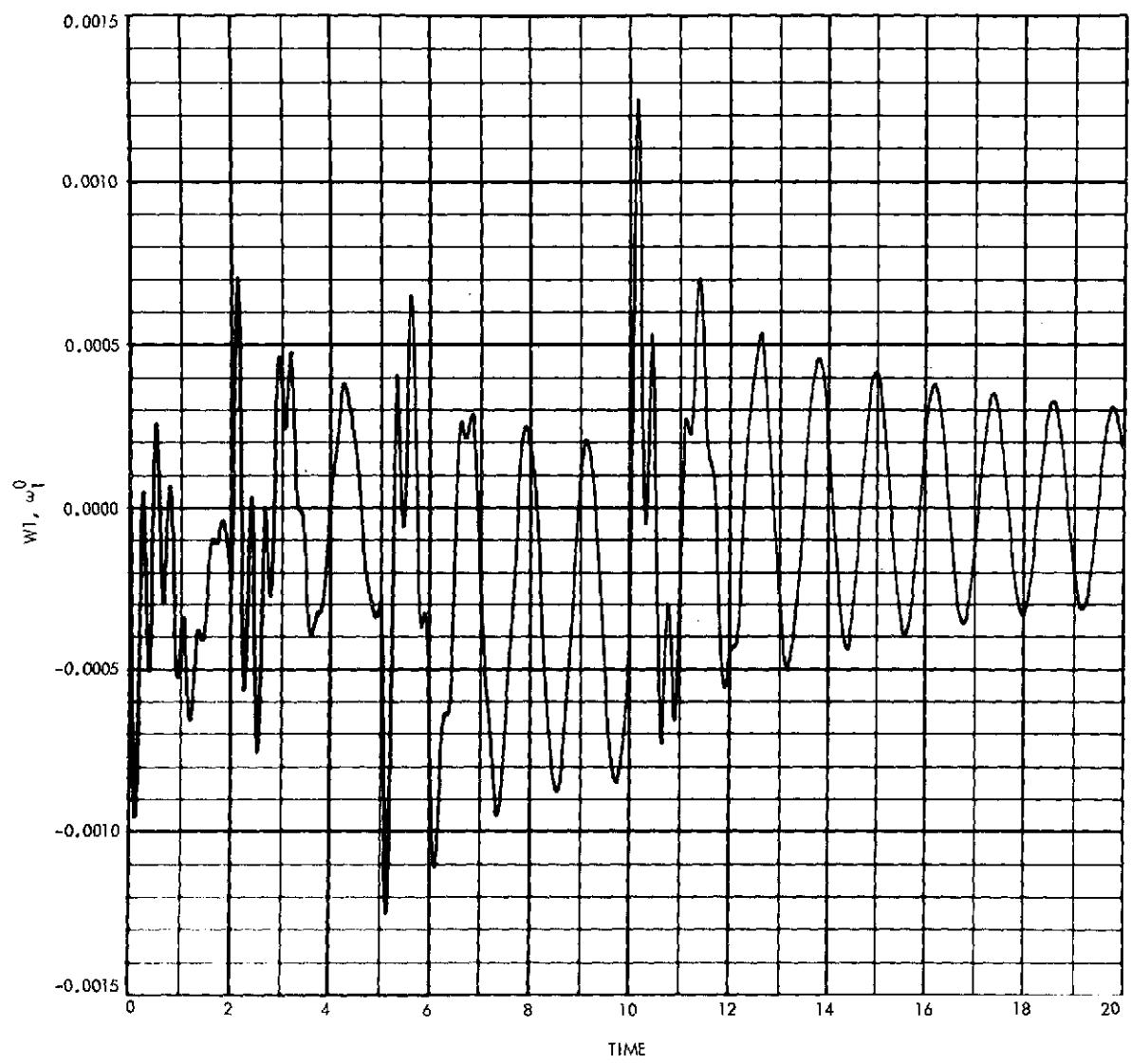


Fig. 14. Results of Case 2 simulation

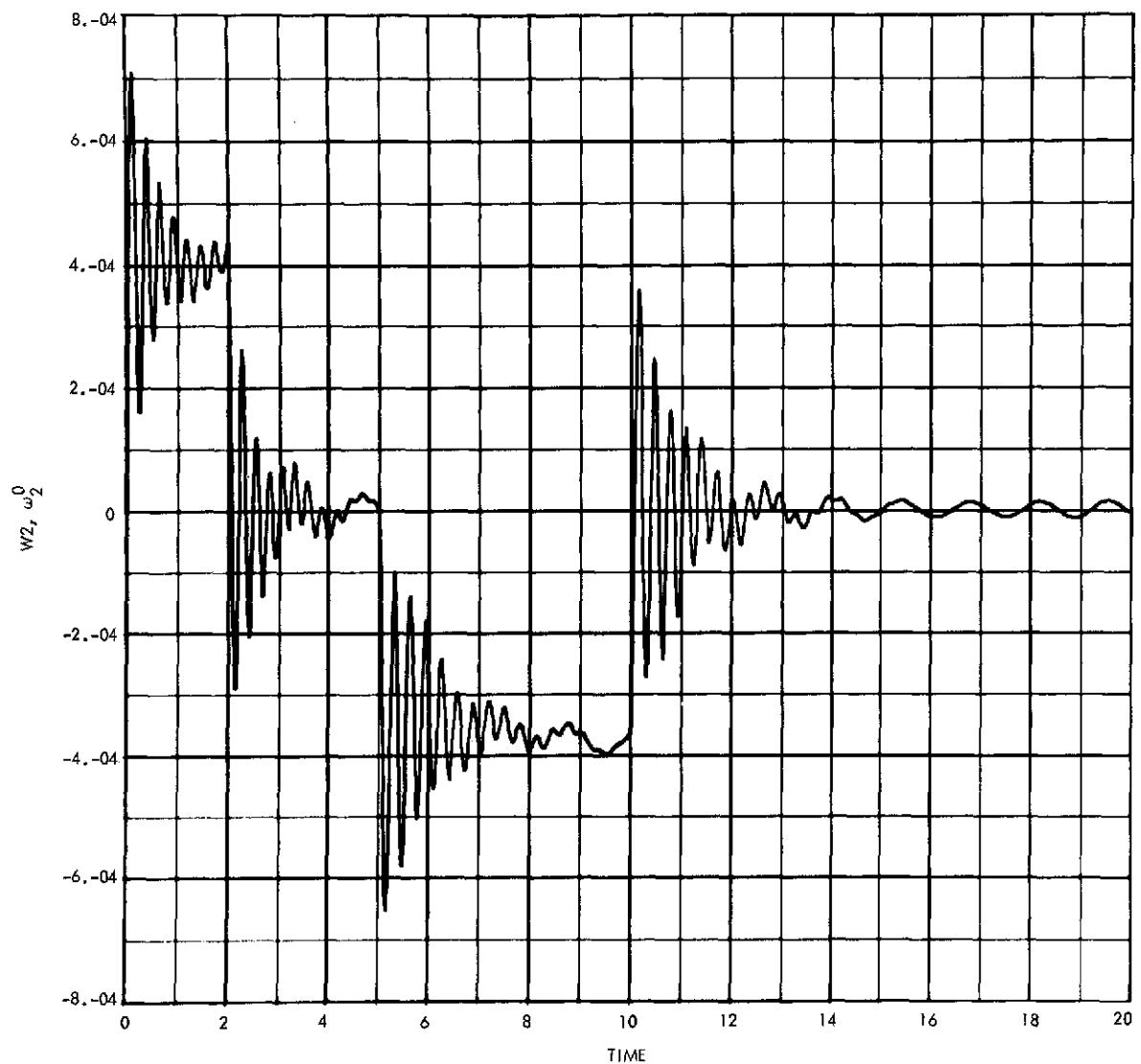


Fig. 14 (contd)

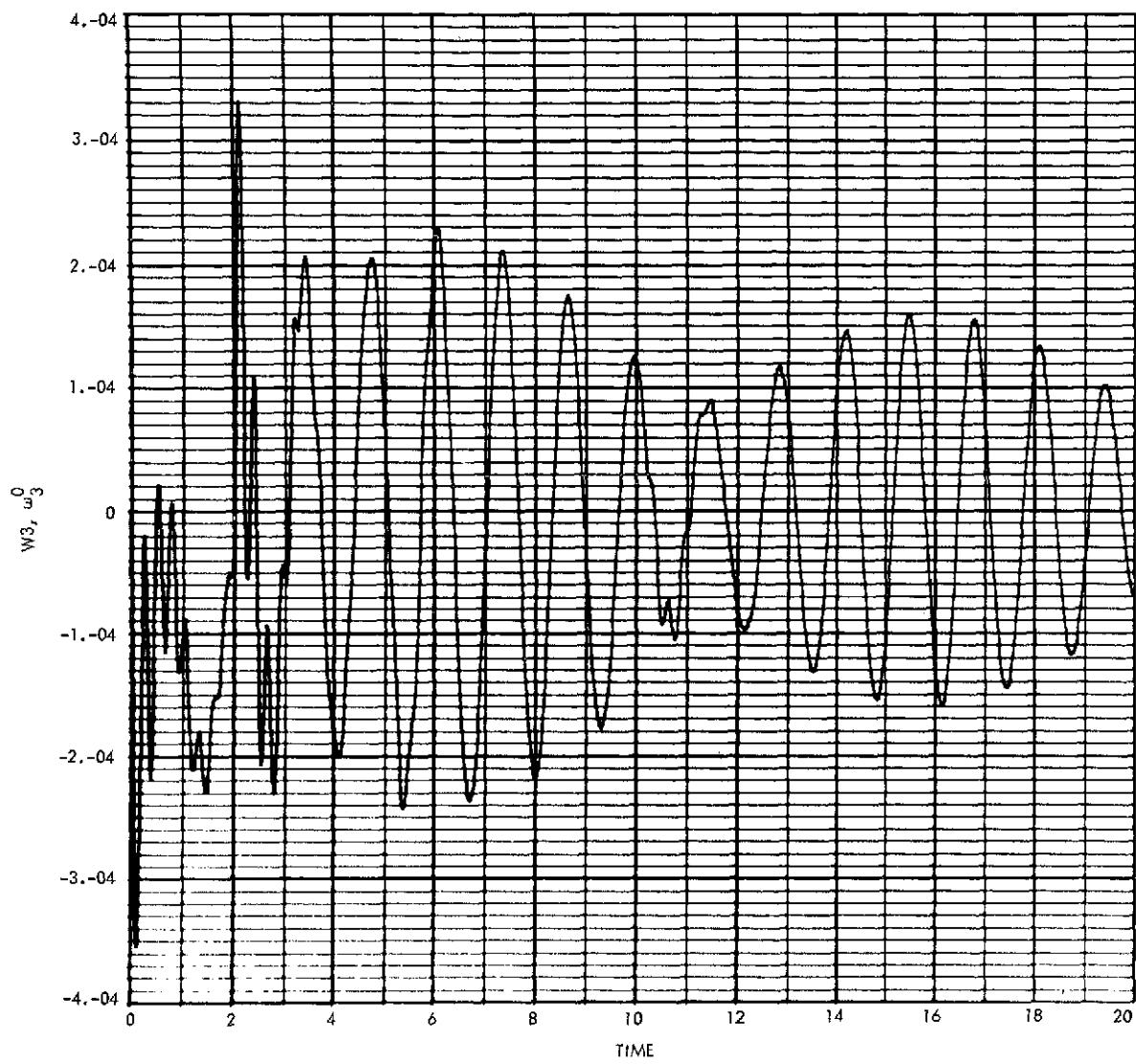


Fig. 14 (contd)

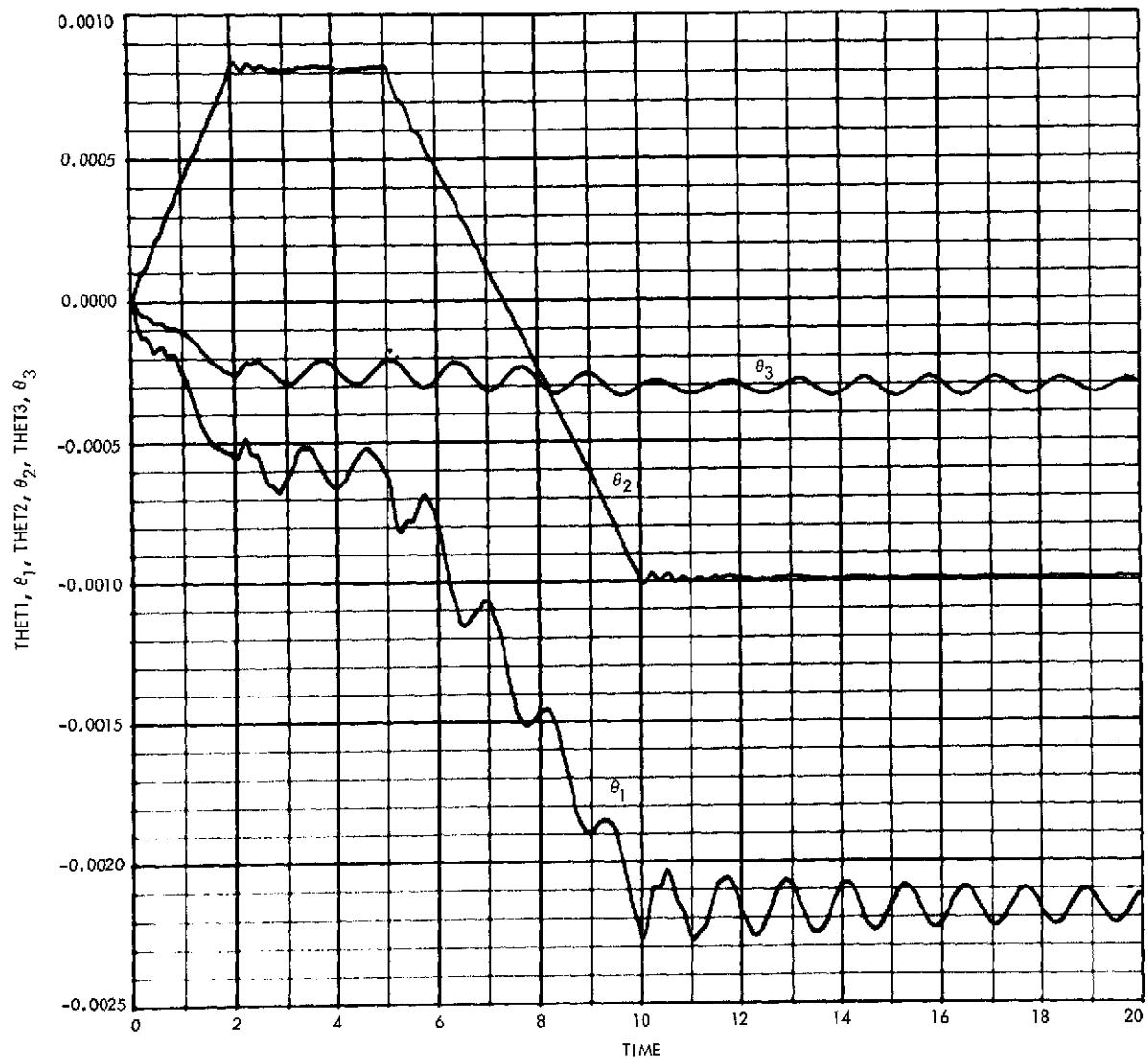


Fig. 14 (contd)

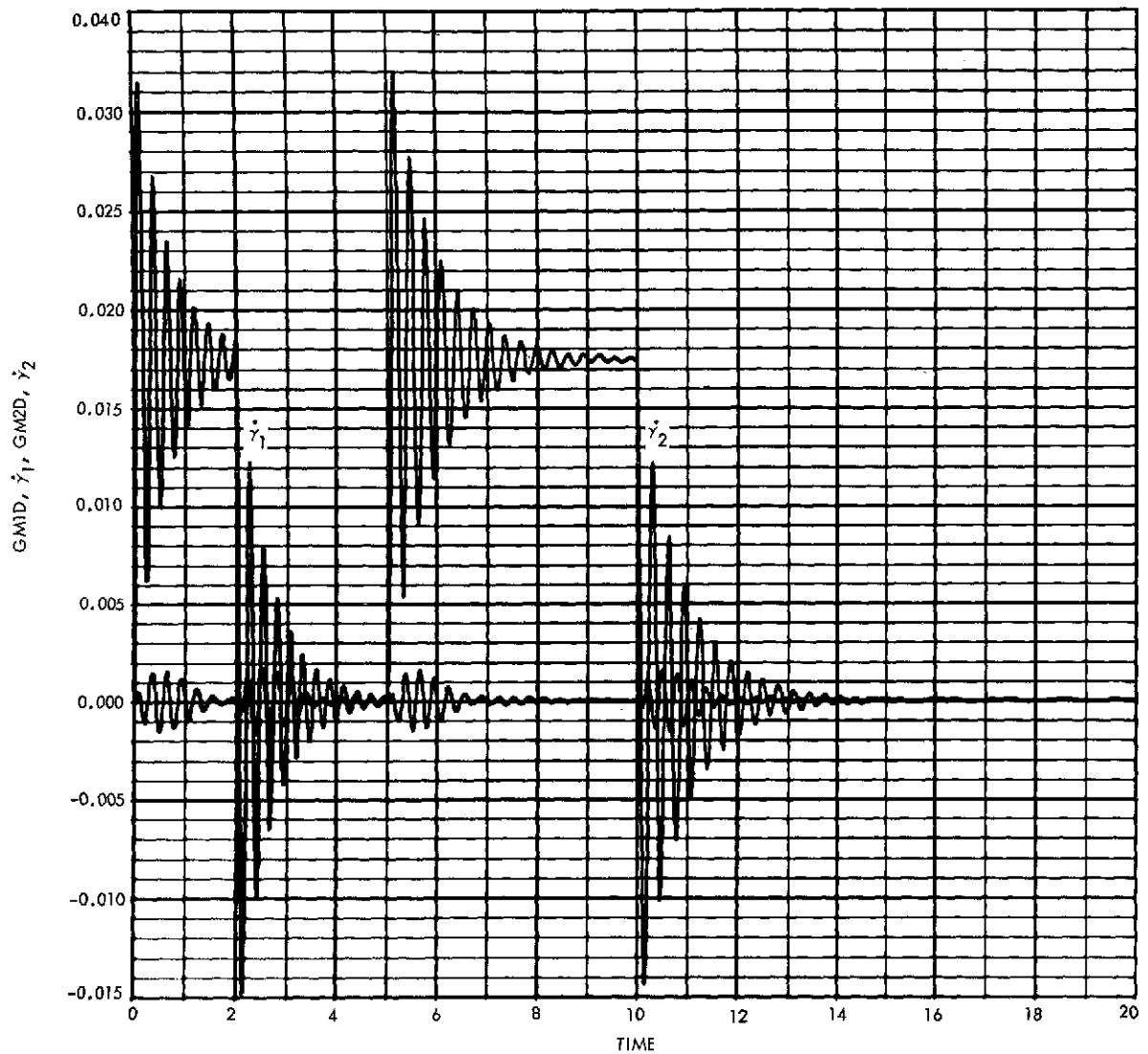


Fig. 14 (contd)

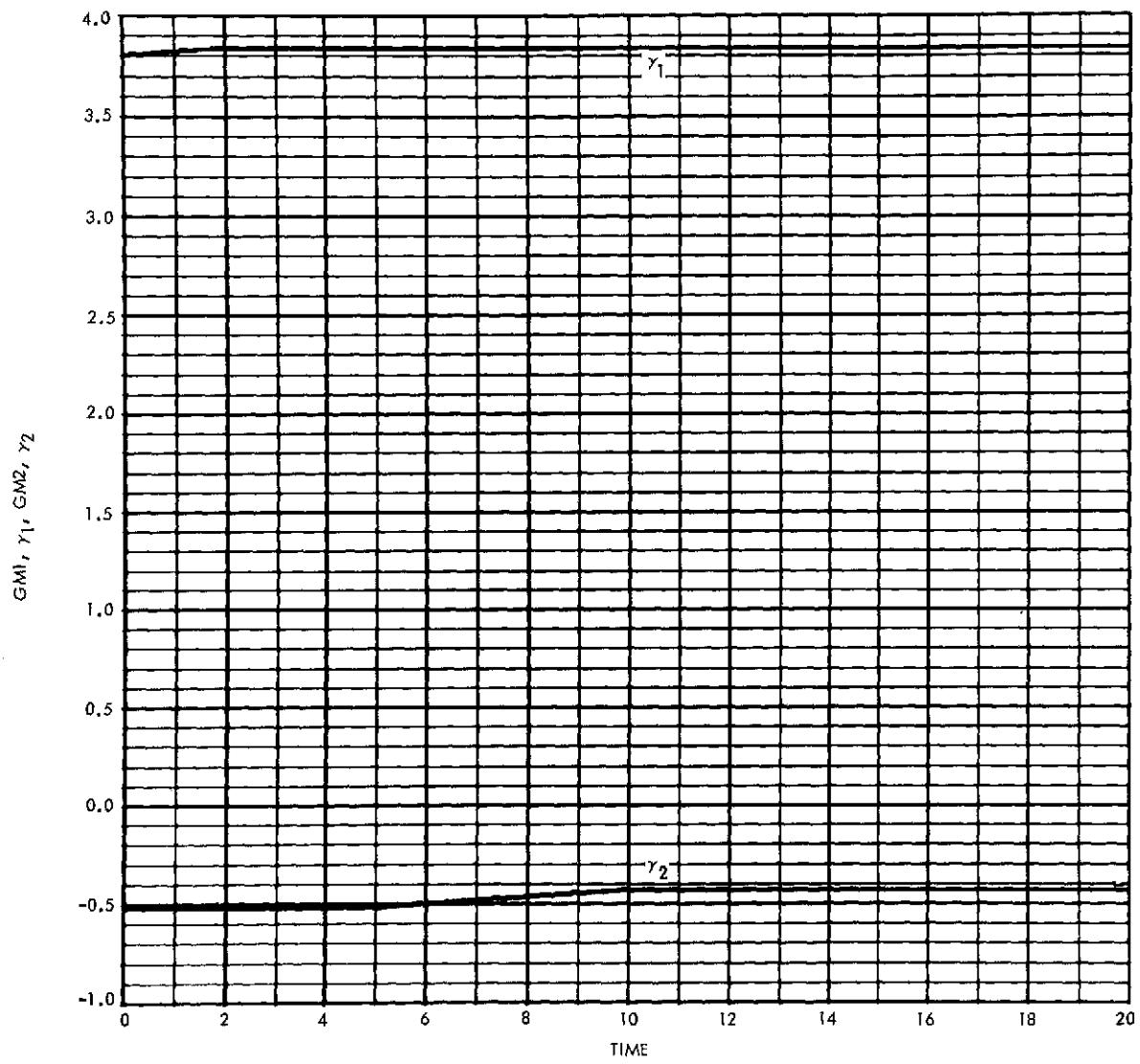


Fig. 14 (contd)

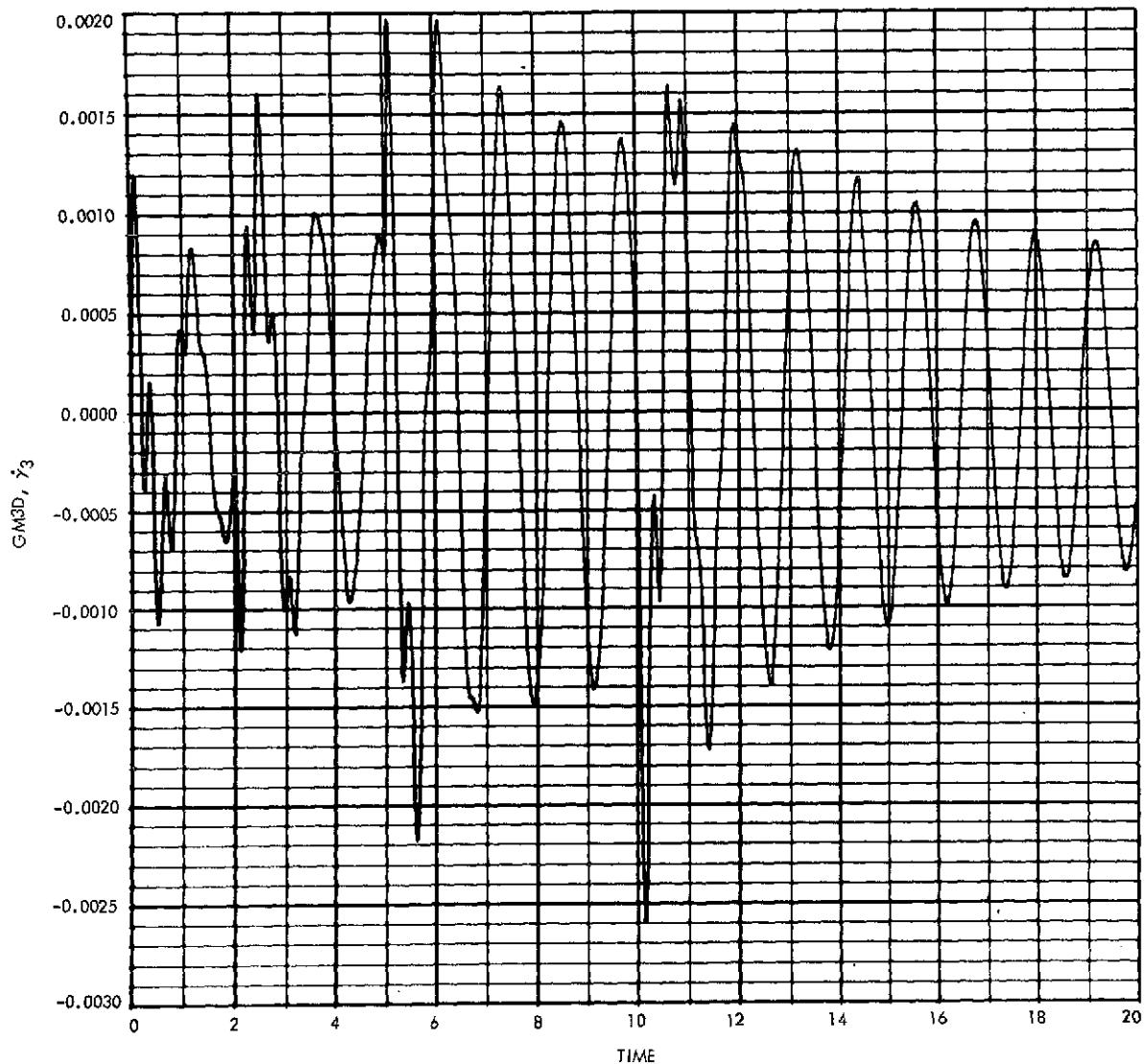


Fig. 14 (contd)

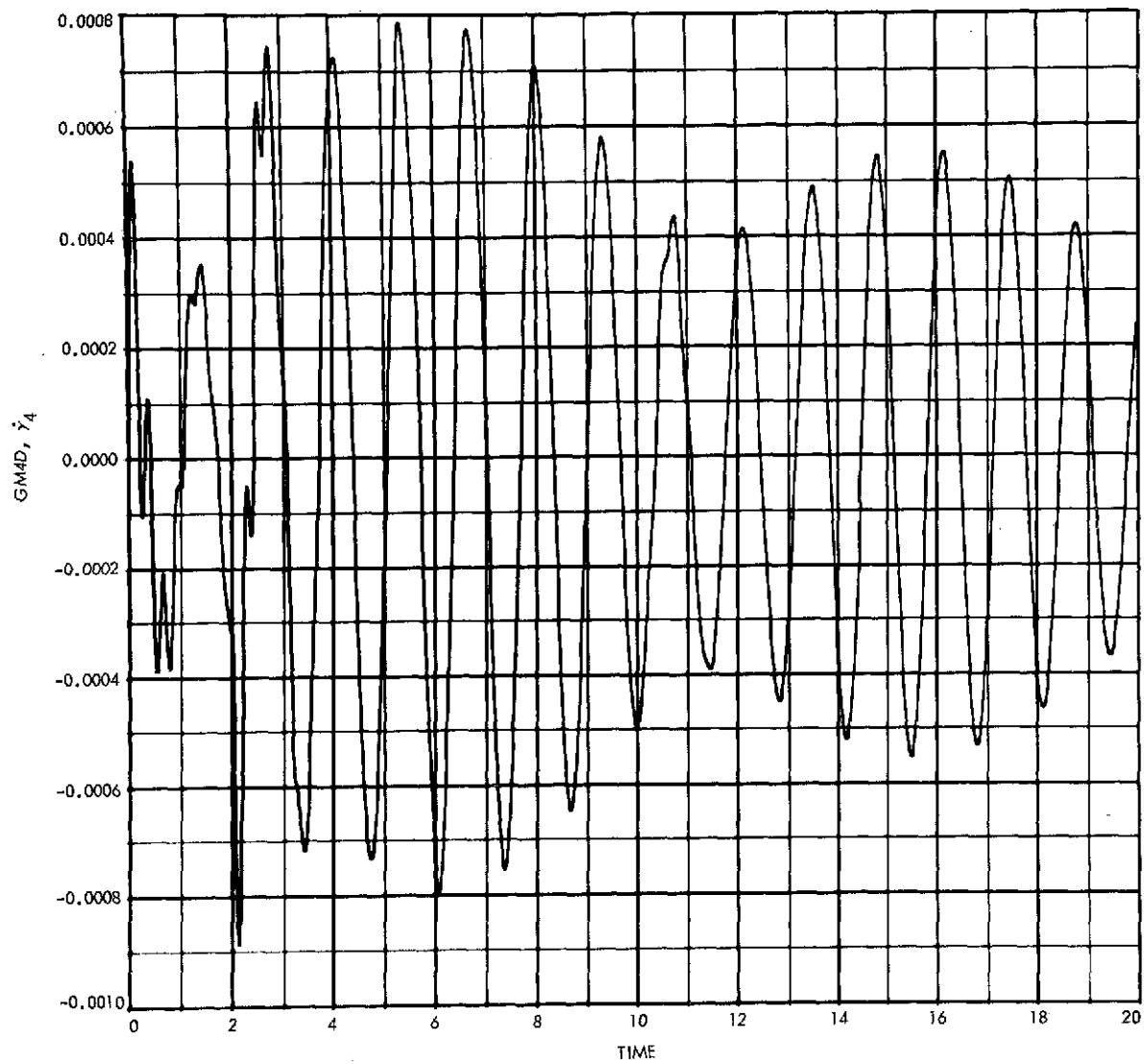


Fig. 14 (contd)

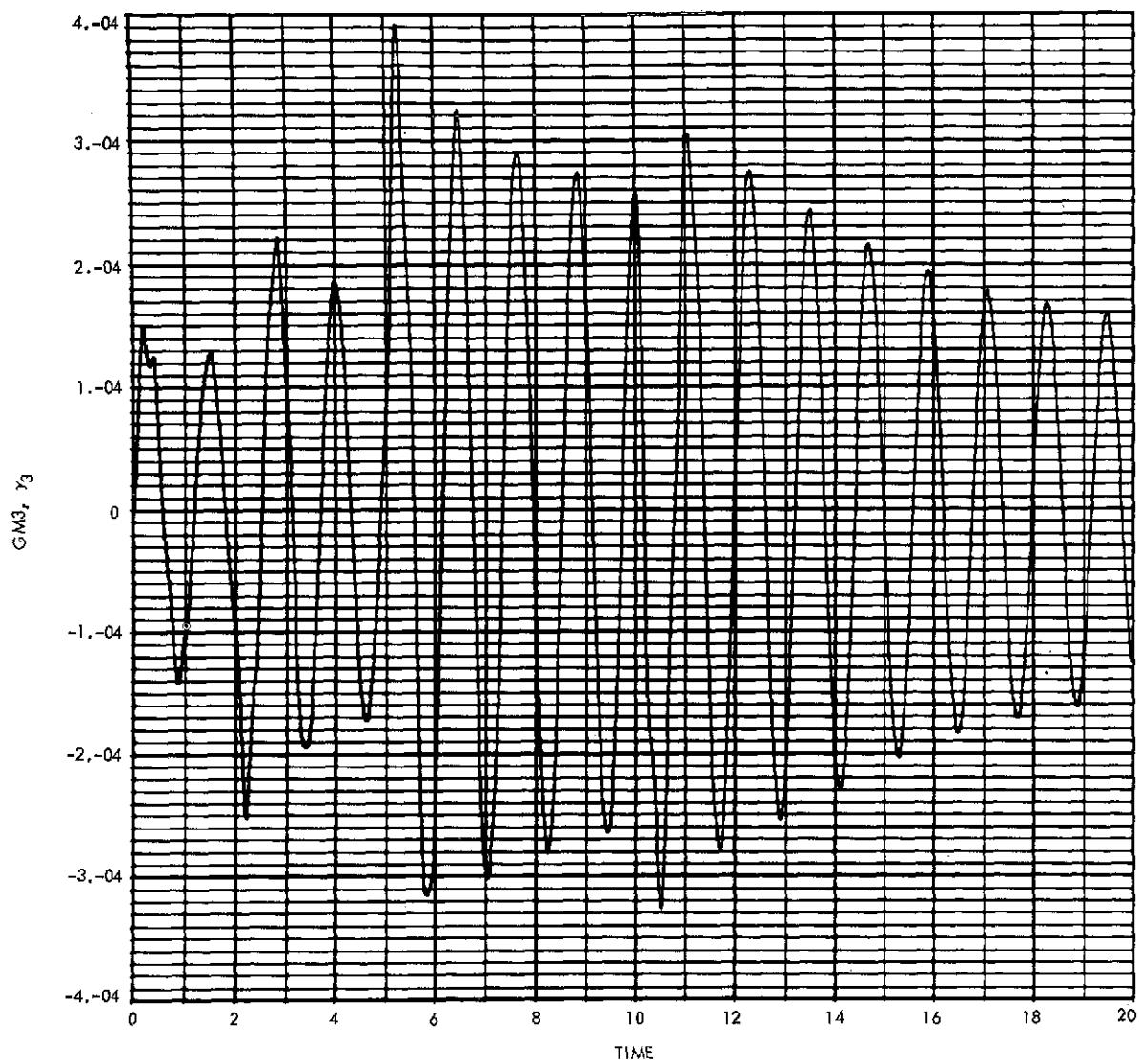


Fig. 14 (contd)

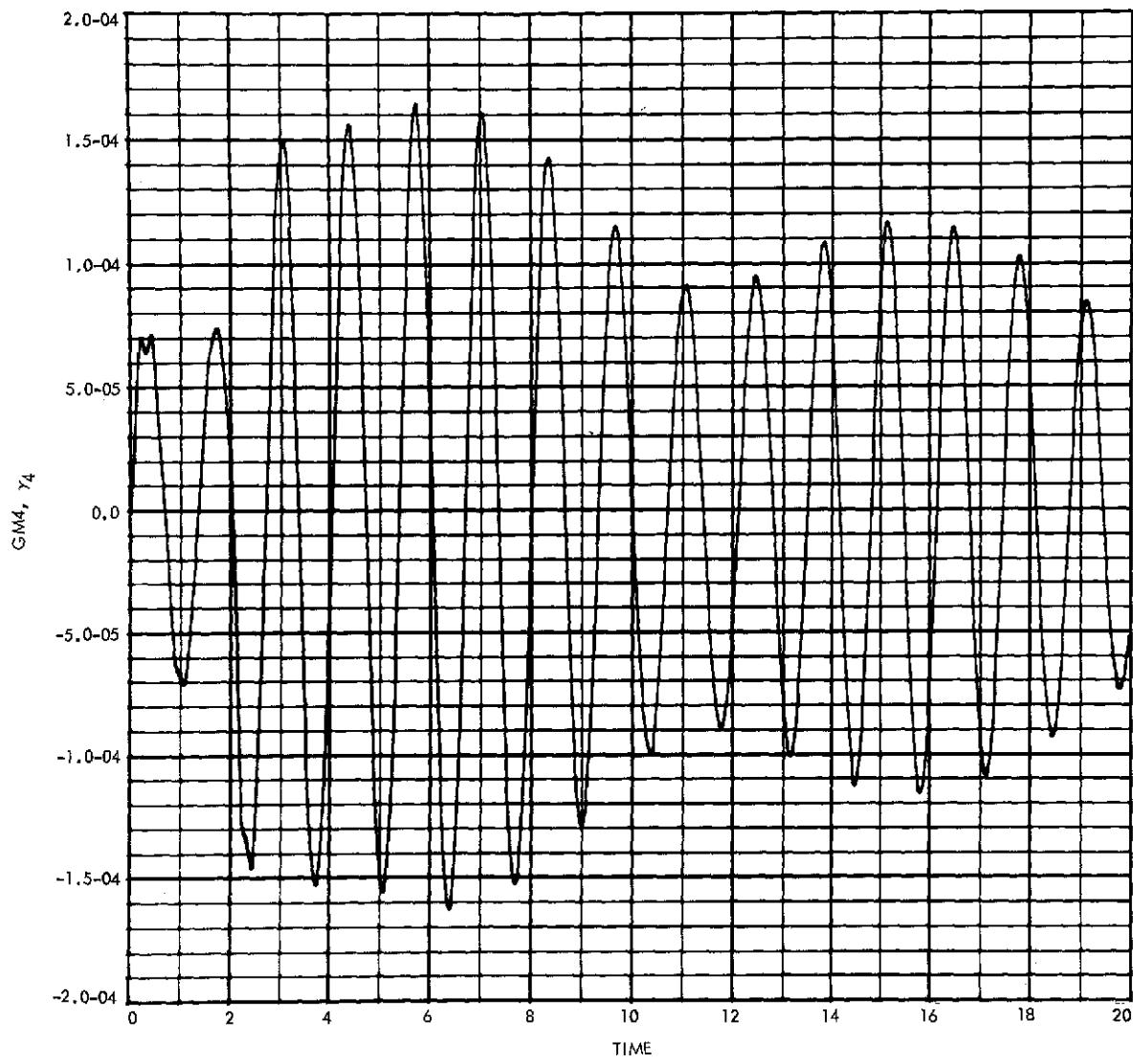


Fig. 14 (contd)

In any case, Eqs. (5) can be rewritten in the form

$$a_{00}\dot{\omega}^0 + \sum_{k \in \mathcal{P}} a_{0k}\ddot{\gamma}_k = \sum_{k \in \mathcal{B}} E_k \quad (8a)$$

and

$$a_{i0}\dot{\omega}^0 + \sum_{k \in \mathcal{P}} a_{ik}\ddot{\gamma}_k = g^{iT} C^{i0} \sum_{k \in \mathcal{P}} \varepsilon_{ik} E_k + \tau_i \quad (i \in \mathcal{P}) \quad (8b)$$

The next step is the linearization of Eq. (8) in the variables ω^0 , γ_k and their time derivatives. To this end, we expand each symbol in Eq. (8) which may involve such variables into three parts, the first being free of these variables (indicated by overbar), the second being linear in the variables (indicated by overcaret), and the third containing terms above the first degree in the variables (indicated by three dots). In particular, we write (for any i, k)

$$C^{i0} = \bar{C}^{i0} + \hat{C}^{i0} + \dots \quad (9a)$$

$$\tau_i = \bar{\tau}_i + \hat{\tau}_i + \dots \quad (9b)$$

$$a_{ik} = \bar{a}_{ik} + \hat{a}_{ik} + \dots \quad (9c)$$

$$E_k = \bar{E}_k + \hat{E}_k + \dots \quad (9d)$$

and then determine explicit expressions for the new barred and caret symbols from the definitions of a_{ik} (Defs. 32, 34, 35), E_k (Def. 36), and the expansion for C^{i0} (Appendix A). In terms of the symbols introduced in Eq. (9), the linearized form of Eq. (8) becomes

$$\bar{a}_{00}\dot{\omega}^0 + \sum_{k \in \mathcal{P}} \bar{a}_{0k}\ddot{\gamma}_k = \sum_{k \in \mathcal{B}} (\bar{E}_k + \hat{E}_k) \quad (10a)$$

$$(i \in \mathcal{P}) \quad \bar{a}_{i0}\dot{\omega}^0 + \sum_{k \in \mathcal{P}} \bar{a}_{ik}\ddot{\gamma}_k = g^i \sum_{k \in \mathcal{P}} \varepsilon_{ik} [\bar{C}^{i0}(\bar{E}_k + \hat{E}_k) + \hat{C}^{i0}\bar{E}_k] + \bar{\tau}_i + \hat{\tau}_i \quad (10b)$$

In deriving explicit expressions for \bar{C}^{i0} and \hat{C}^{i0} , we may begin by recognizing that for any C^{rj} , where r and j are labels of *adjacent* bodies,

$$\bar{C}^{rj} = \bar{C}^{jr} = U = 3 \times 3 \text{ identity matrix}$$

and

$$\hat{C}^{rj} = -\gamma_r \tilde{g}_r, \quad (r > j)$$

$$\hat{C}^{jr} = \gamma_r \tilde{g}^r = (\hat{C}^{rj})^T$$

It is clear that \hat{C}^{i0} must then be given by

$$\hat{C}^{i0} = -\sum_{s \in \mathcal{P}} \varepsilon_{si} \tilde{g}^s \gamma_s, \quad \hat{C}^{0i} = (\hat{C}^{i0})^T = \sum_{s \in \mathcal{P}} \varepsilon_{si} \tilde{g}^s \gamma_s$$

Also, $\bar{C}^{i0} = U$. In general,

$$(r, j \in \mathcal{B}) \quad \hat{C}^{rj} = \sum_{s \in \mathcal{P}} (\varepsilon_{sj} - \varepsilon_{sr}) \tilde{g}^s \gamma_s \quad (11a)$$

and

$$\bar{C}^{rj} = U \quad (11b)$$

From Def. 36, \bar{E}_k and \hat{E}_k are given by

$$\bar{E}_k = \bar{C}^{0k} [\bar{T}^k + \sum_{j \in \mathcal{B}} \tilde{D}^{kj} \bar{C}^{kj} \bar{F}^j] = \bar{T}^k + \sum_{j \in \mathcal{B}} \tilde{D}^{kj} \bar{F}^j \quad (12a)$$

$$\begin{aligned} \hat{E}_k &= \bar{C}^{0k} [\hat{T}^k + \sum_{j \in \mathcal{B}} \tilde{D}^{kj} \hat{C}^{kj} \bar{F}^j + \sum_{j \in \mathcal{B}} \tilde{D}^{kj} \bar{C}^{kj} \hat{F}^j] + \hat{C}^{0k} [\bar{T}^k + \sum_{j \in \mathcal{B}} \tilde{D}^{kj} \bar{C}^{kj} \bar{F}^j] \\ &= \hat{T}^k + \sum_{j \in \mathcal{B}} \tilde{D}^{kj} \hat{C}^{kj} \bar{F}^j + \sum_{j \in \mathcal{B}} \tilde{D}^{kj} \hat{F}^j + \hat{C}^{0k} [\bar{T}^k + \sum_{j \in \mathcal{B}} \tilde{D}^{kj} \bar{F}^j] \end{aligned} \quad (12b)$$

Of course, the most important result of the complete linearization is the absence of any time-varying elements in \bar{a}_{00} , \bar{a}_{0k} , \bar{a}_{i0} , and \bar{a}_{ik} . These *constant* coefficients now allow the square matrix on the left side of (5) to be inverted once and for all, in advance of numerical integration.

It would remain then to determine \bar{T}^k , \hat{T}^k , \bar{F}^k , \hat{F}^k , $\bar{\tau}_i$, and $\hat{\tau}_i$ for the specific system under study and, together with the relations in (11) and (12), to carry out the computations implied by (10). However, in constructing a subroutine to perform these computations, it was found to be much easier to directly manipulate the combined form:

$$\bar{a}_{00}\dot{\omega}^0 + \sum_{k \in \mathcal{P}} \bar{a}_{0k}\ddot{\gamma}_k = \sum_{k \in \mathcal{B}} \hat{\bar{E}}_k \quad (13a)$$

$$\bar{a}_{i0}\dot{\omega}^0 + \sum_{k \in \mathcal{P}} \bar{a}_{ik}\ddot{\gamma}_k = g^{iT} \hat{\bar{C}}^{i0} \sum_{k \in \mathcal{P}} \epsilon_{ik} \hat{\bar{E}}_k + \hat{\tau}_i \quad (i \in \mathcal{P}) \quad (13b)$$

where

$$\begin{aligned} \hat{\bar{E}}_k &= \hat{\bar{C}}^{0k} [\hat{\bar{T}}^k + \sum_{j \in \mathcal{B}} \tilde{D}^{kj} \hat{\bar{C}}^{kj} \hat{\bar{F}}^j] \\ \hat{\bar{C}}^{i0} &= \bar{C}^{i0} + \hat{C}^{i0} \\ \hat{\bar{C}}^{0k} &= \bar{C}^{0k} + \hat{C}^{0k} \\ \hat{\bar{T}}^k &= \bar{T}^k + \hat{T}^k \\ \hat{\bar{F}}^j &= \bar{F}^j + \hat{F}^j \\ \hat{\bar{\tau}}_i &= \bar{\tau}_i + \hat{\tau}_i \end{aligned}$$

By dealing only with the variables $\hat{\bar{E}}_k$, $\hat{\bar{C}}^{0k}$, etc., rather than separating them into the parts \bar{E}_k , \hat{E}_k , etc., and explicitly forming all the separate terms in (10) and (12), the computation becomes much more straightforward and efficient even though some second-order terms in the linearized variables are retained.

One further form of the system equations, which goes beyond strict linearization in order to simplify and speed the computations, is the following:

$$\bar{a}_{00}\dot{\omega}^0 + \sum_{k \in \mathcal{P}} \bar{a}_{0k}\ddot{\gamma}_k = \sum_{k \in \mathcal{B}} E'_k \quad (14a)$$

$$\bar{a}_{i0}\dot{\omega}^0 + \sum_{k \in \mathcal{P}} \bar{a}_{ik}\ddot{\gamma}_k = g^{iT} \sum_{k \in \mathcal{P}} \epsilon_{ik} E'_k + \hat{\tau}_i \quad (14b)$$

where

$$E'_k = \bar{C}^{0k} [\hat{\bar{T}}^k + \sum_{j \in \mathcal{B}} \tilde{D}^{kj} \hat{\bar{C}}^{kj} \hat{\bar{F}}^j] = \hat{\bar{T}}^k + \sum_{j \in \mathcal{B}} \tilde{D}^{kj} \hat{\bar{F}}^j$$

Equations (14) obviate the computation of any body-to-body coordinate transformation matrices, \bar{C}^{ri} , at all.

Subroutines which provide solutions to both Eqs. (13) and (14) will be described in the next section.

B. Subroutines MBDYTL and MBDYL

The subroutines MBDYTL and MBDYL provide the solutions to Eqs. (13) and (14) of the preceding section, respectively. As in the case of MBDY, they each may be exercised by either of two call statements, the first of which initializes the program with the system constants.

Initializing Call Statements

```
CALL MBDYTL(NH,H,MB,MA,PB,PA,G,PI)
```

or

```
CALL MBDYL(NH,H,MB,MA,PB,PA,G,PI)
```

Except for the call statement name, the statements are exactly the same as that for MBDY. All the arguments are defined exactly as given in Section IIE.

To compute the angular accelerations $\dot{\omega}^0, \ddot{\gamma}_1, \dots, \ddot{\gamma}_n$ the subroutine MBDYTL (or MBDYL) must be entered at every integration interval. This is accomplished by executing the dynamic call statement.

Dynamic Call Statement

```
CALL MRATE(NH,TH,TB,TA,FB,FA,GM,CMD,GMDD,WO,WDOT)
```

Again, the dynamic call statement arguments are the same as those defined for MBDY in Section IIE, with the following *exceptions*:

$TH(i)$ = array of hinge torques, $\hat{T}_i; i \in \mathcal{P}$

$TB(j)$ = array of elements of $\hat{T}^0; j = 1, 2, 3$

$TA(i,j)$ = array of elements of $\hat{T}^i; i \in \mathcal{P}, j = 1, 2, 3$

$FB(j)$ = array of elements of $\hat{F}^0; j = 1, 2, 3$

$FA(i,j)$ = array of elements of $\hat{F}^i; i \in \mathcal{P}, j = 1, 2, 3$

Even though the arguments GMD and WO are not used by either MBDYTL or MBDYL, and GM is not used by MBDYL, these "dummy" arguments are retained in the call statements for the sake of uniformity and to allow easy interchange of subroutines if desired.

The Fortran "type" and storage allocation statements which must appear in the main program that calls MBDYTL (or MBDYL) are identical to those required for MBDY.

Required Specification Statements

```
INTEGER NH, H(n), PI(n)
```

```
REAL MB(7), MA(n,7), PB(n,3), PA(n,n,3), G(n,3), TH(n),
   TB(3), TA(n,3), FB(3), FA(n,3), GM(n), GMD(n),
   GMDD(n), WO(3)
```

```
DOUBLE PRECISION WDOT(n + 3)
```

Also, the statement

PARAMETER $Q = n$

must appear *within* the subroutine MBDYTL (or MBDYL) to allocate the proper amount of internal storage. Complete listings of the MBDYTL and MBDYL programs are given in Appendix B.

C. Application of MBDYTL and MBDYL to the Five-Body System

1. Small platform maneuver simulation. We may now proceed to apply the linearized subroutines to the same vehicle dynamics/control configuration examined in Section II.F. The vehicle is shown in Fig. 4 and the control system in Fig. 5. Expressions previously given for T^0 , τ_1 , τ_2 , τ_3 , and τ_4 may now be re-interpreted as describing \hat{T}^0 , $\hat{\tau}_1$, $\hat{\tau}_2$, $\hat{\tau}_3$, and $\hat{\tau}_4$, respectively, since they contain no second-order or higher terms in the variables ω^0 , γ_k and their derivatives. In fact, in order to again simulate Case 2, i.e., the small platform maneuver from $\gamma_1 = 218$ deg to $\gamma_1 = 220$ deg (at $t = 0$) and $\gamma_2 = -30$ deg to $\gamma_2 = -25$ deg (at $t = 5$ s), it is necessary to replace the statement "CALL MBDY(NH,...)" with the statement "CALL MBDYTL(NH,...)" and to modify the arrays MA(2,i), PA(1,2,j), PA(2,2,j), and G(2,j).

The reason for the required modifications to these arrays stems from Defs. 4 and 12, which require all local, body-fixed coordinate frames to coincide (i.e., $\mathbf{b}_\alpha^k = \mathbf{b}_\alpha^j$ for $\alpha = 1, 2, 3$ and $k, j \in \mathcal{B}$) when the system is in its nominal static, i.e., when $\gamma_k = 0$ for all $k \in \mathcal{P}$. When the routines MBDYTL and MBDYL are used, the variables γ_k are, by definition, required to remain very close to the nominal state $\gamma_k = 0$. Otherwise, the linearization process is meaningless. It is clear, then, that to simulate Case 2 with the linearized programs, the new linearized variables $\hat{\gamma}_k$ ($k = 1, 4$) must be redefined in terms of the old γ_k such that, for example,

$$\hat{\gamma}_1 = \gamma_1 - 219 \text{ deg}$$

$$\hat{\gamma}_2 = \gamma_2 + 27.5 \text{ deg}$$

$$\hat{\gamma}_3 = \gamma_3$$

$$\hat{\gamma}_4 = \gamma_4$$

And now it is also clear that bodies \mathbf{b}_1 and \mathbf{b}_2 have different orientations in their own body-fixed frames, implying different inertia matrices I^1 , I^2 and different components for the vectors \mathbf{p}^{10} , \mathbf{p}^{12} , \mathbf{p}^{21} , \mathbf{g}^1 , \mathbf{g}^2 in their respective frames, $\{\mathbf{b}^1\}$ and $\{\mathbf{b}^2\}$.

In this particular vehicle configuration, the platform hub, \mathbf{b}_1 , has symmetrical mass properties about its axis of rotation, which in turn is aligned to one of the body-fixed coordinate axes. Thus, I^1 , and therefore MA(1,j), $j = 1 - 7$, remains unchanged, as does \mathbf{p}^{10} (PA(1,1,j), $j = 1 - 3$) and \mathbf{g}^1 (G(1,j), $j = 1 - 3$). However, it is apparent that \mathbf{p}^{12} , \mathbf{p}^{21} , as well as \mathbf{g}^2 , and I^2 must change, since they were previously defined for $\gamma_1 = 0$, $\gamma_2 = 0$ and now must reflect the configuration defined by $\hat{\gamma}_1 = 0$, $\hat{\gamma}_2 = 0$ ($\gamma_1 = 219$ deg, $\gamma_2 = -27.5$ deg).

The components of a vector described in the old frame fixed in ϵ_2 can be related to its components in the new frame by the sequence of rotations γ_1 and γ_2 , as follows:

$$\begin{Bmatrix} v_1^N \\ v_2^N \\ v_3^N \end{Bmatrix} = \begin{bmatrix} c\gamma_1 & -s\gamma_1c\gamma_2 & -s\gamma_1s\gamma_2 \\ s\gamma_1 & c\gamma_1c\gamma_2 & c\gamma_1s\gamma_2 \\ 0 & -s\gamma_2 & c\gamma_2 \end{bmatrix} \begin{Bmatrix} v_1^0 \\ v_2^0 \\ v_3^0 \end{Bmatrix} = C^{N0} \begin{Bmatrix} v_1^0 \\ v_2^0 \\ v_3^0 \end{Bmatrix}$$

$$(s\gamma_1 = \sin \gamma_1, c\gamma_1 = \cos \gamma_1, \text{etc.})$$

Therefore, the components of \mathbf{g}^2 in a simulation linearized about $\gamma_1 = 219^\circ$, $\gamma_2 = -27.5^\circ$ deg become:

$$\begin{Bmatrix} g_1^2 \\ g_2^2 \\ g_3^2 \end{Bmatrix} = C^{N0} \begin{Bmatrix} -1 \\ 0 \\ 0 \end{Bmatrix} = \begin{Bmatrix} -c\gamma_1 \\ -s\gamma_1 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 0.77715 \\ 0.62932 \\ 0 \end{Bmatrix}$$

Similarly, the components of \mathbf{p}^{21} become

$$\begin{Bmatrix} p_1^{21} \\ p_2^{21} \\ p_3^{21} \end{Bmatrix} = C^{N0} \begin{Bmatrix} 0 \\ -0.22 \\ 0.20 \end{Bmatrix} = \begin{Bmatrix} 0.22s\gamma_1c\gamma_2 - 0.20s\gamma_1s\gamma_2 \\ 0.20c\gamma_1s\gamma_2 - 0.22c\gamma_1c\gamma_2 \\ 0.22s\gamma_2 + 0.20c\gamma_2 \end{Bmatrix} = \begin{Bmatrix} -0.18092 \\ 0.22342 \\ 0.07582 \end{Bmatrix}$$

The vector \mathbf{p}^{12} only changes due to the γ_1 rotation, i.e.,

$$\begin{Bmatrix} p_1^{12} \\ p_2^{12} \\ p_3^{12} \end{Bmatrix} = \begin{bmatrix} c\gamma_1 & -s\gamma_1 & 0 \\ s\gamma_1 & c\gamma_1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{Bmatrix} 0 \\ -0.10 \\ -0.75 \end{Bmatrix} = \begin{Bmatrix} 0.1s\gamma_1 \\ -0.1c\gamma_1 \\ -0.75 \end{Bmatrix} = \begin{Bmatrix} -0.06293 \\ 0.07772 \\ -0.75 \end{Bmatrix}$$

Finally, the modified value of I^2 is given by

$$I^2 = C^{N0} \begin{bmatrix} 4.85 & 0.41 & -0.07 \\ 0.41 & 2.20 & 0.54 \\ -0.07 & 0.54 & 5.50 \end{bmatrix} (C^{N0})^T = \begin{bmatrix} 3.52 & 1.25 & -0.75 \\ 1.25 & 3.79 & 0.73 \\ -0.75 & 0.73 & 5.24 \end{bmatrix}$$

Having determined these changes in the system configuration description, the appropriate values of $MA(2,i)$, $i = 1 - 6$, $PA(1,2,j)$, $PA(2,2,j)$, $G(2,j)$ may be entered to produce the simulation program of Fig. 15. Notice in Fig. 15 that it was also necessary to change $IC1$ and $IC2$ to reflect proper initial values of the new variables $\hat{\gamma}_1$ and $\hat{\gamma}_2$ (i.e., $\hat{\gamma}_1(0) = -1^\circ$, $\hat{\gamma}_2(0) = -2.5^\circ$; these variables correspond to $\gamma_1(0) = 218^\circ$, $\gamma_2(0) = -30^\circ$). Using this program and the MBDYTL subroutine, Case 2 was simulated again, and the results are shown in Fig. 16. They may be directly compared to the dynamic responses of Fig. 14. While there are some very slight differences visible in the angular rate responses, the overall result is virtually identical with that obtained using MBDY, and is obtained in *one third the time used by MBDY* (MBDY used 140 s, MBDYTL 48 s on the Univac 1108). Further, the substitution of MBDYL for MBDYTL gives exactly the same results as those shown in Fig. 16, since no *external* forces or torques were applied to the system, and calculates them in only 42 s.

Obviously, the gain in computational speed when the linearized routines are used is quite significant, even for this relatively simple example. In Ref. 7, a 17-degree-of-freedom system is simulated using MBDYTL exclusively, since it gave a speed advantage over MBDY of about 25:1 with more than adequate solution accuracy.

```

PROGRAM S-BODY SYSTEM SIMULATION USING MBDYTL --- CASE 2
  INTEGER NH,PI(4),H14
  ARRAY MA(7),MA(4,7),PB(4,31),PA(4,4,3),G(4,3),TR(3),TA(4,3)
  ARRAY FB(3),FA(4,31),TH14,GM(4),GMD(4),GMDD(4),W0(3)
  DOUBLE PRECISION WDDOT(7)
  DATA H/0.,1,0,3/PI/0.,0,0/0
  DATA MB/115.,316.,440.,14.,-14.,34.,6,410./
  DATA MA(1,1)/.35/MA(1,2)/.35/MA(1,7)/.87
  DATA MA(2,1)/.3,.52/MA(2,21)/.73/MA(2,31)/.5,.24/MA(2,4)/-1.25/MA(2,5)/.75/
  DATA MA(2,6)/-.73/MA(2,7)/.57,.5/
  DATA MA(4,1)/.27,.2/MA(4,2)/.2/MA(4,3)/.27,.2/MA(4,7)/10.7/
  DATA PB(1,3)/-1.5/PB(1,2)/-1.2/
  DATA PA(1,1,31)/-.7i/PB(1,2,11)/-.66233/PA(1,2,21)/.07772/PA(1,2,31)/-.75/
  DATA PA(2,2,11)/-.18032/PA(2,2,21)/.22342/PA(2,2,31)/.07582/
  DATA PA(4,4,21)/3.3/
  DATA G(1,31)/1./G(2,1)/.22715/612,21/.62932/G(3,1)/1./G(4,31)/1./
  CONSTANT TM=.020,.08=.005,TR=2.0
  CONSTANT TFINAL=2E.,CLKTIME=110C.
  CONSTANT L1=.23,-.21,-.21=.31
  CONSTANT SLRATC=.01745323
  CONSTANT PHI2=0.,THET2=0.,PSIZE=C.
  CONSTANT IC1=-1.,IC2=-2.5
  CONSTANT W1=U.,W2=0.,W3=0.
  CONSTANT K1=3500.,K2=3500.,K3=2000.,K4=2000.
  CONSTANT S1=20.,S2=20.,S3=10.,S4=10.

INITIAL
  NH=4
  RD=57.295773E $ DR=1./RD $ GM1C=IC1/RD $ GM2C=IC2/RD
  GM1I=IC1/RD $ GM2I=IC2/RD
  CALL MBDYTL(NH,H,MB,MA,PB,PA,G,PI,
  PZI,PAI,PBI,PC1=4CKIINITL,PHIZ,THET2,PSIZ)

END
DYNAMIC
  IF TIME.GE.TFINAL GO TO S1
DERIVATIVE BODY5
  VARIABLE TIME=0,
  CINTERVAL CT=C
  XERROR W1=1.E-6,W2=1.E-6,W3=1.E-6,GM3=1.E-4
  SI=CLK CLKTIME
  OUTPUT 50.*W10+W20+W30,W1,W2,W3,THET1,THET2,THET3...
  IN1,IN2,IN3,GM1,GM2,GM3,GM4,PL1,PL2
  PREPAR W1,W2,W3,THET1,THET2,THET3,GM10,GM20...
  GM1,GM2,GM30,GM40,GM3,GM4
NOSORT
  W0(1)=W1 $ W0(2)=W2 $ W0(3)=W3
  PL1=RD*GM1 $ PL2=RD*GM2
  GM(11)=GM1 $ GM(21)=GM2 $ GM(31)=GM3 $ GM(41)=GM4
  GMD(11)=GM10 $ GMD(21)=GM20 $ GMD(31)=GM30 $ GMD(41)=GM40
COMMENT
COMMENT CELESTIAL-SENSOR-MEASURED ANGLES
COMMENT
  D1,D2=HCK1/MATRIX,PZC,PAC,PBC,PC01
  NX0,NY0,NZ0=HCK1/I10B,E,,C,,1,,D1,D2
  LX0,LY0,LZ0=HCK1/I10B,1,,0,,0,,D1,D2
  THE11=ATAN2(NY0,NZ0) $ THE12=ATAN2(-NX0,NZ0)
  THE13=ATAN2(-LY0,LX0)

```

Fig. 15. Case 2 simulation using MBDYTL

```

COMMENT
COMMENT      RATE FEEDBACK SIGNAL AND SWITCHING AMP. DYNAMICS
COMMENT
    IN1=-THET1-TR*W1   $   IN2=-THET2-TR*W2   $   IN3=-THET3-TR*W3
    AMP1=SWAMPITIME+TM,DB+IN1)
    AMP2=SWAMPITIME+TM,DB+IN2)
    AMP3=SWAMPITIME+TM,DB+IN3)
COMMENT
COMMENT      PLATFORM POSITION COMMANDS
COMMENT
    GM1CD=SLRAT*(1,-STEP12.,TIME1) $ GM1C=INTEG(GM1CD,GM1CI)
    GM2CD=SLRAT*(STEP15.,TIME1-STEP10.,TIME1)$ GM2C=INTEG(GM2CD,GM2CI)
COMMENT
COMMENT      REFERENCE BODY EXTERNAL TORQUE
COMMENT      HINGE TORQUES
COMMENT
    TB(1)=L1*AMP1 $ TB(2)=L2*AMP2 $ TB(3)=L3*AMP3
    TH(1)=K1*(GM1-GP1CI-B1*GM1D
    TH(2)=K2*(GM2-GM2CI-B2*GM2D
    TH(3)=K3*GM3-R3*GM3D
    TH(4)=K4*GM4-E4*GM4D
COMMENT
COMMENT      SYSTEM ANGULAR ACCELERATIONS
COMMENT
    CALL MRATEINH(TH,TB,TA,FB,FA,GM,GMD,GMD0,W0,W0DT)
    W10=W0DT(1) $ W20=W0DT(2) $ W30=W0DT(3)
COMMENT
COMMENT      SYSTEM ANGULAR RATES AND POSITIONS
COMMENT
    W1=INTEG(W0DT(1),W1I)
    W2=INTEG(W0DT(2),W2I)
    W3=INTEG(W0DT(3),W3I)
    GM1D=INTEG(W0DT(4),0.) $ GM2D=INTEG(W0DT(5),0.)
    GM3D=INTEG(W0DT(6),0.) $ GM4D=INTEG(W0DT(7),0.)
    GM1=INTEG(GM1D,GM1I) $ GM2=INTEG(GM2D,GM2I)
    GM3=INTEG(GM3D,0.) $ GM4=INTEG(GM4D,0.)
COMMENT
COMMENT      HCK PARAMETER RATES AND POSITIONS
COMMENT
    PZ00,P400,P300,PC00=HCK(HCK,PZ0,P40,P30,PC0,W1,W2,W3)
    PZ0=INTEG(PZ00,PZI)
    P40=INTEG(P400,P4I)
    P30=INTEG(P300,PBI)
    PC0=INTEG(PC00,PCI)
END
END
END
TERMINAL
SI.. CONTINUE
END
END

```

Fig. 15 (contd)

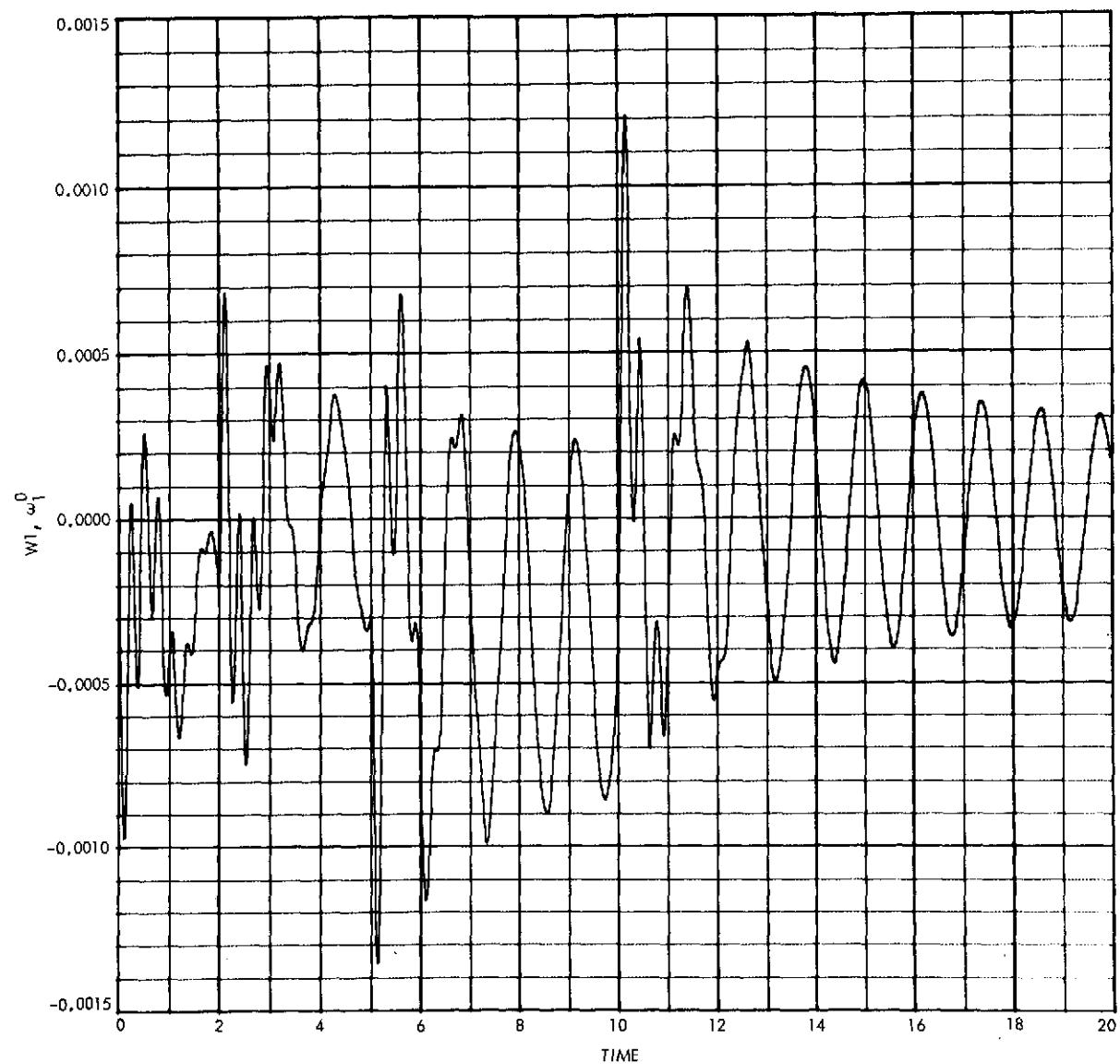


Fig. 16. Results of Case 2 simulation using MBDYTL

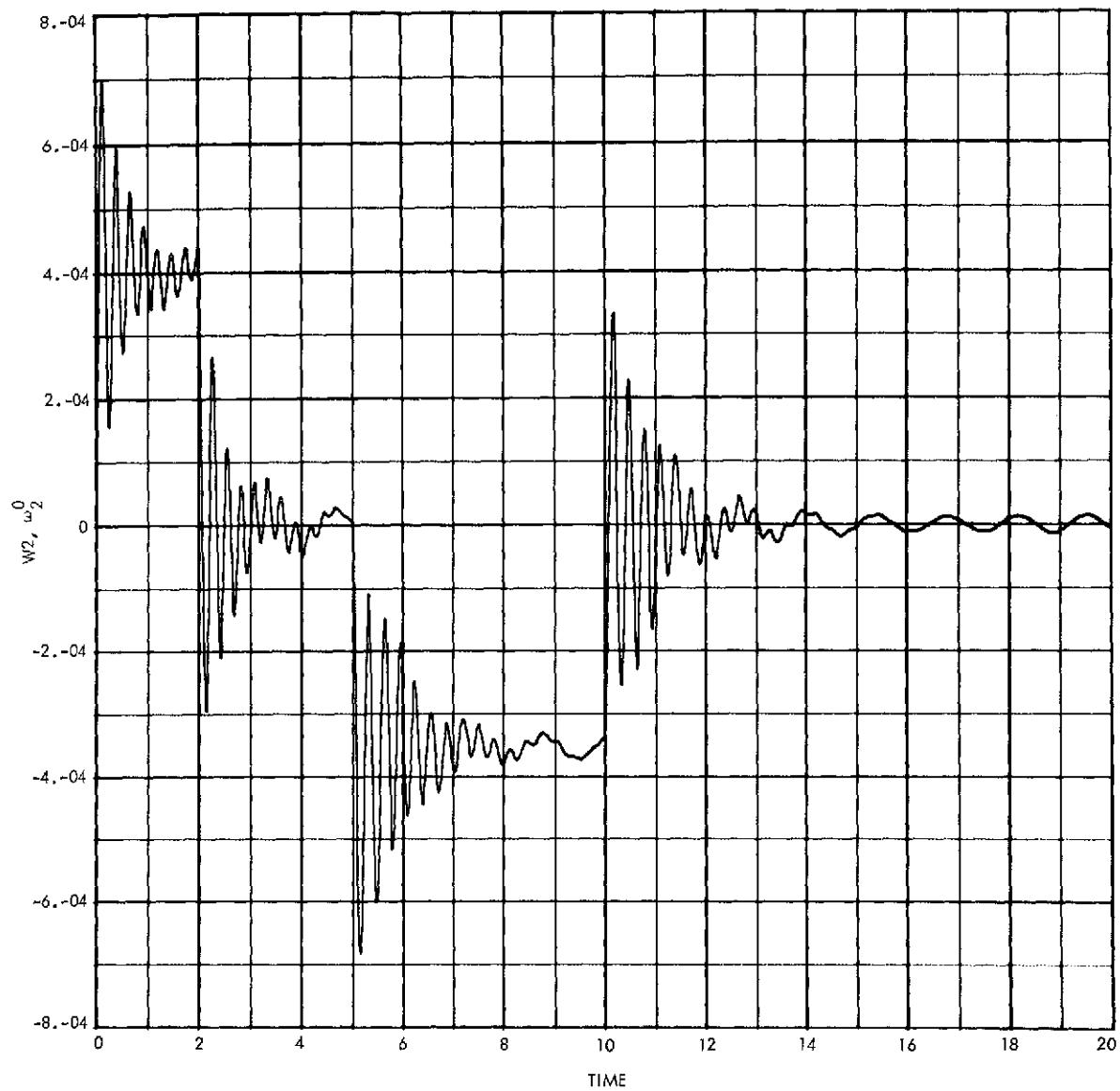


Fig. 16 (contd)

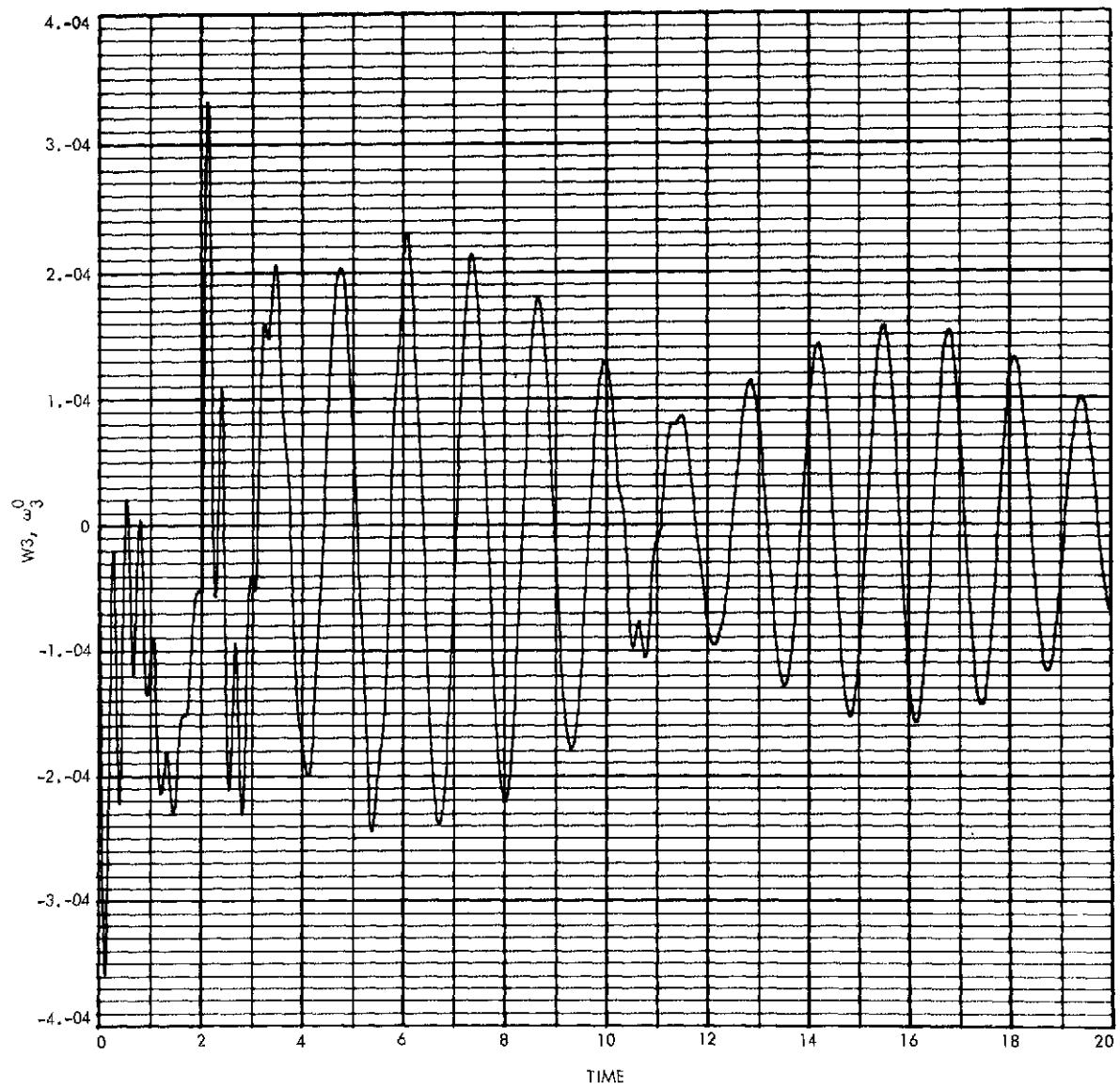


Fig. 16 (contd)

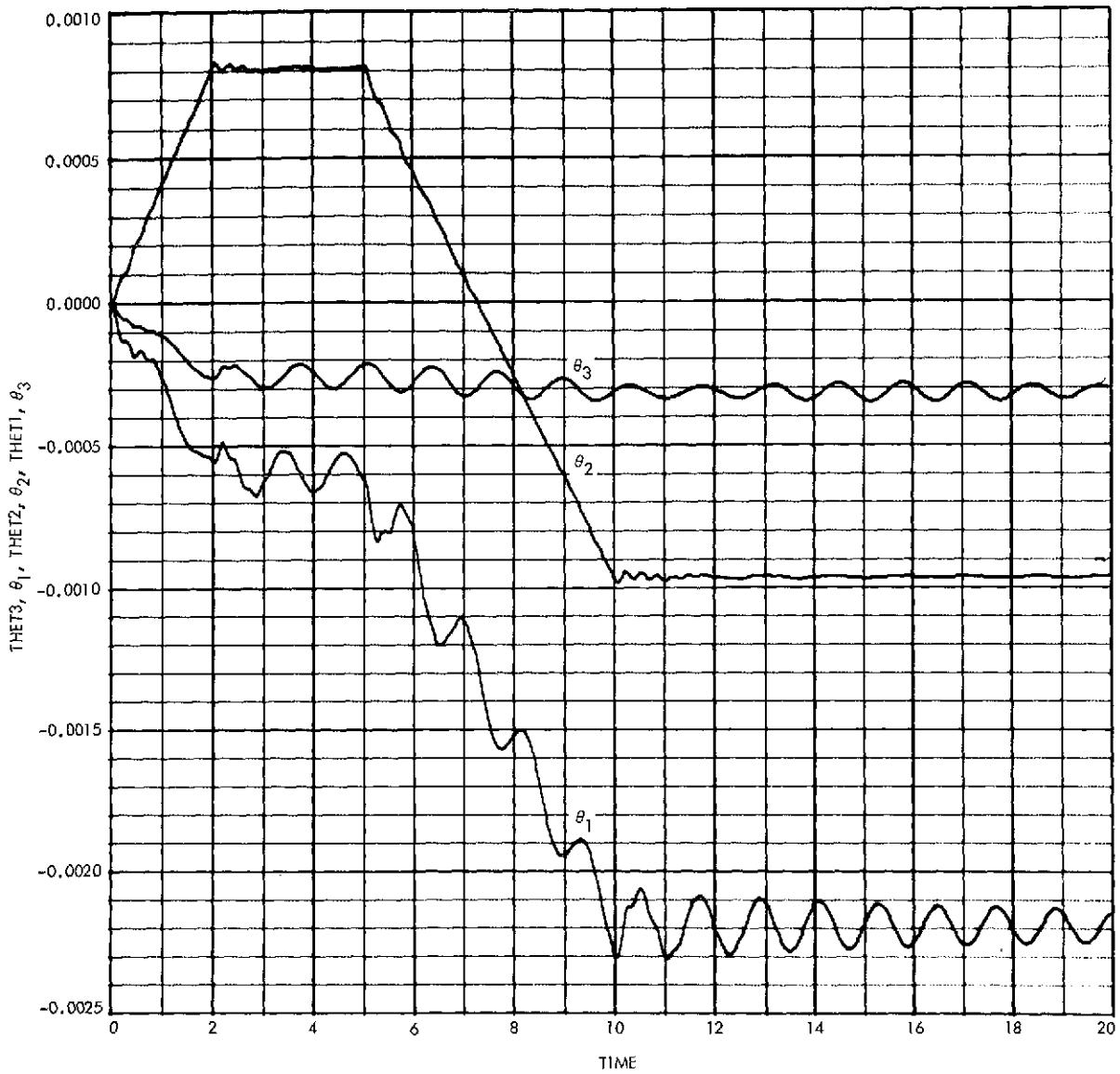


Fig. 16 (contd)

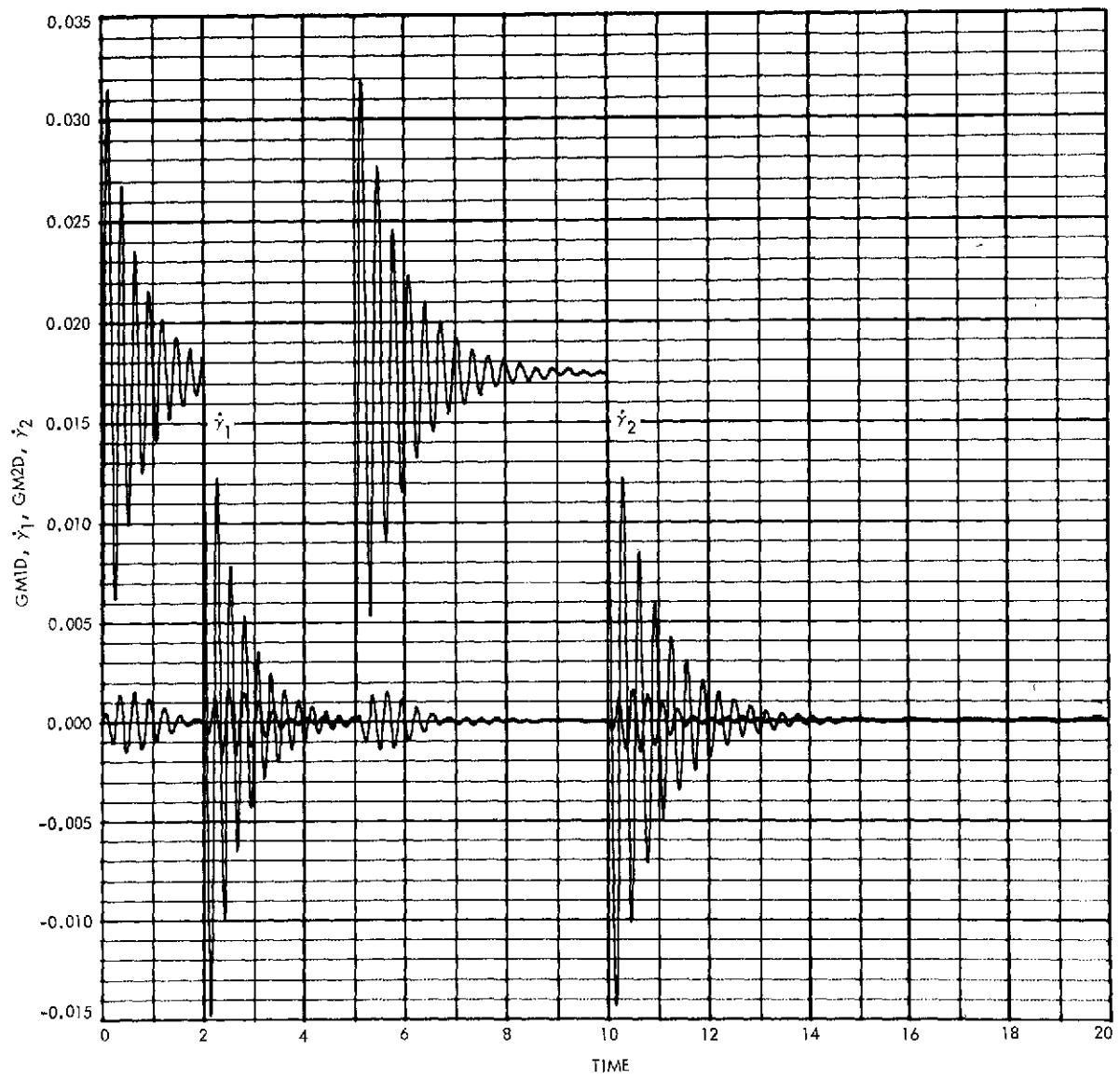


Fig. 16 (contd)

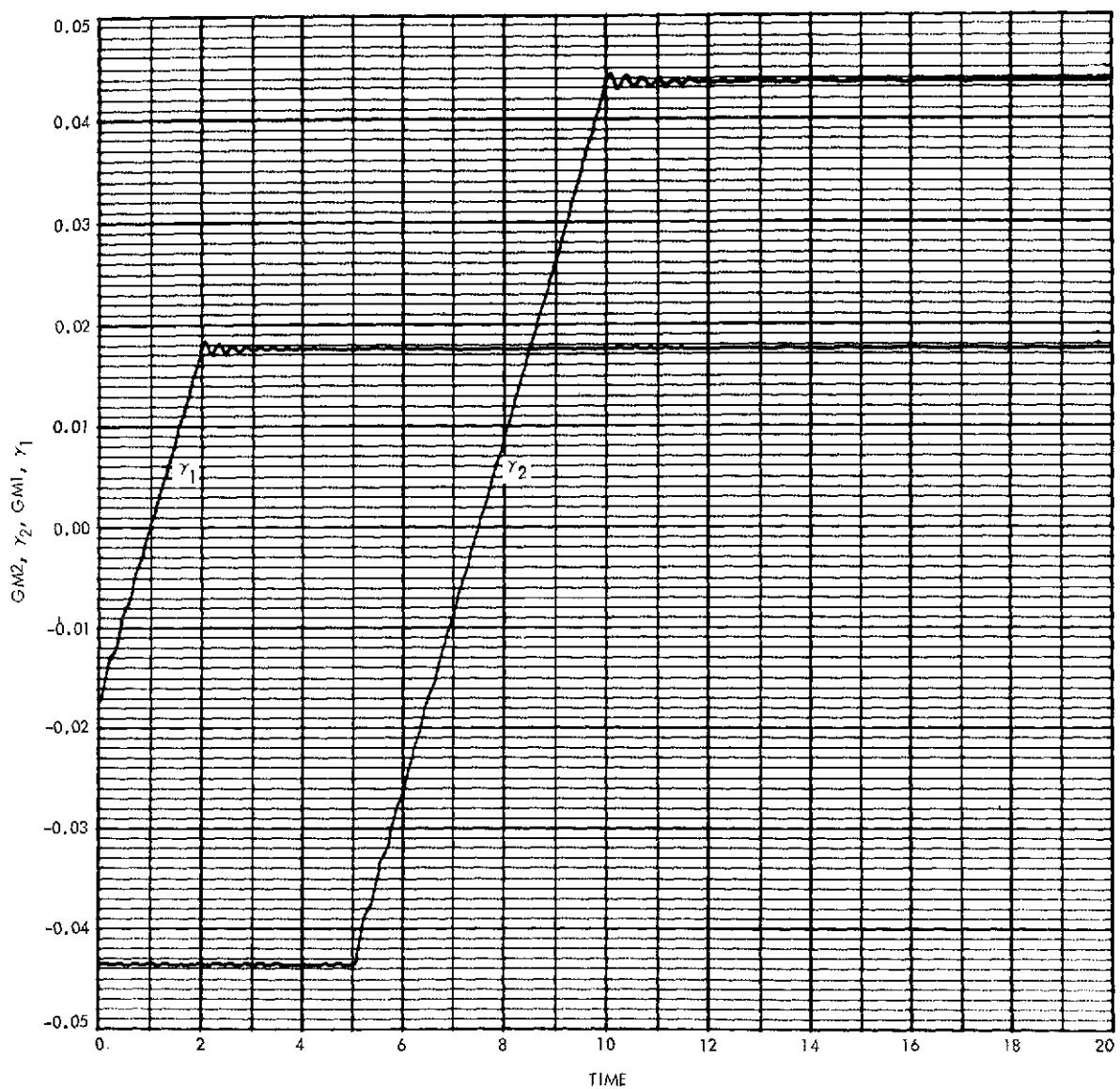


Fig. 16 (contd)

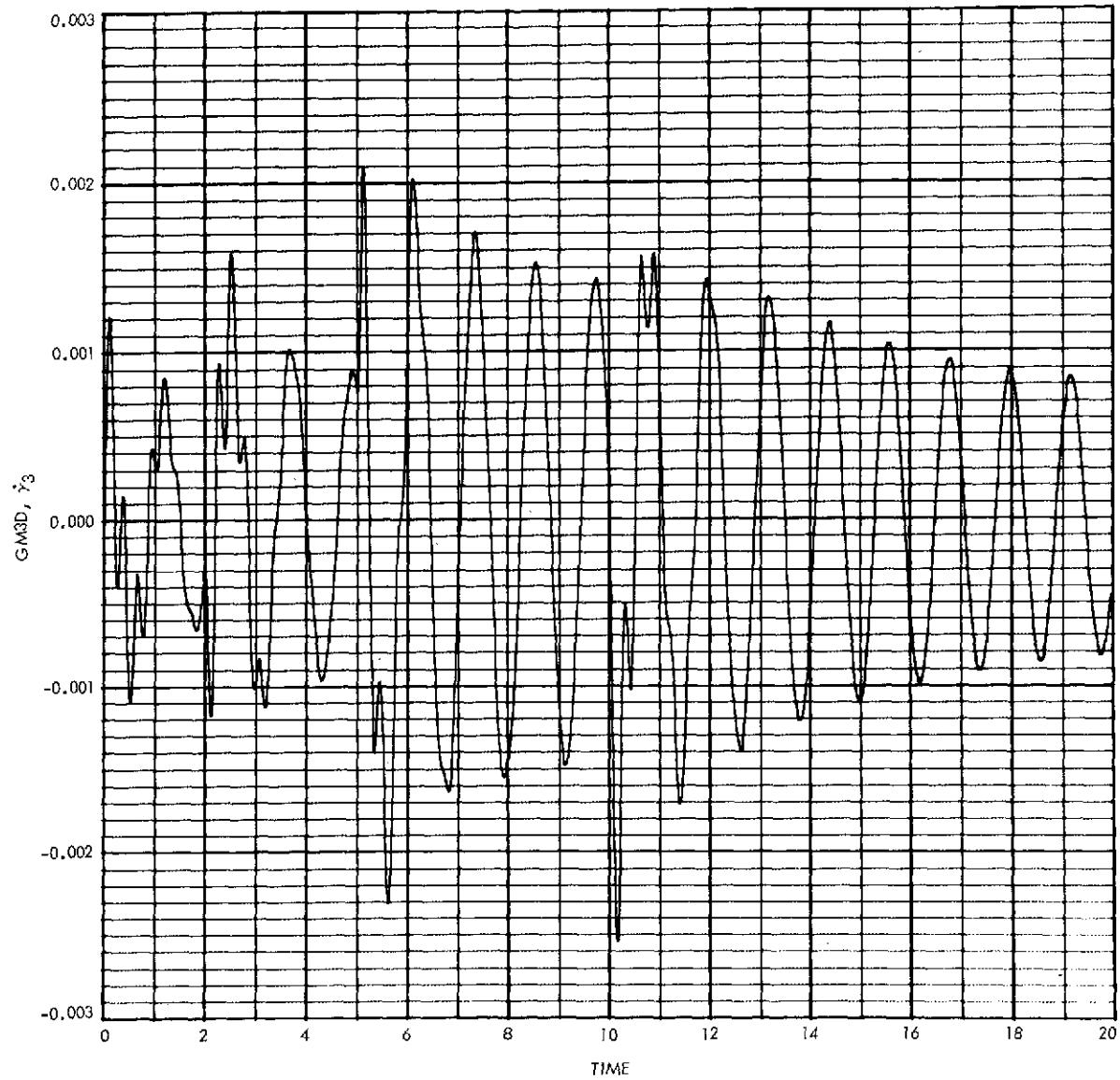


Fig. 16 (contd)

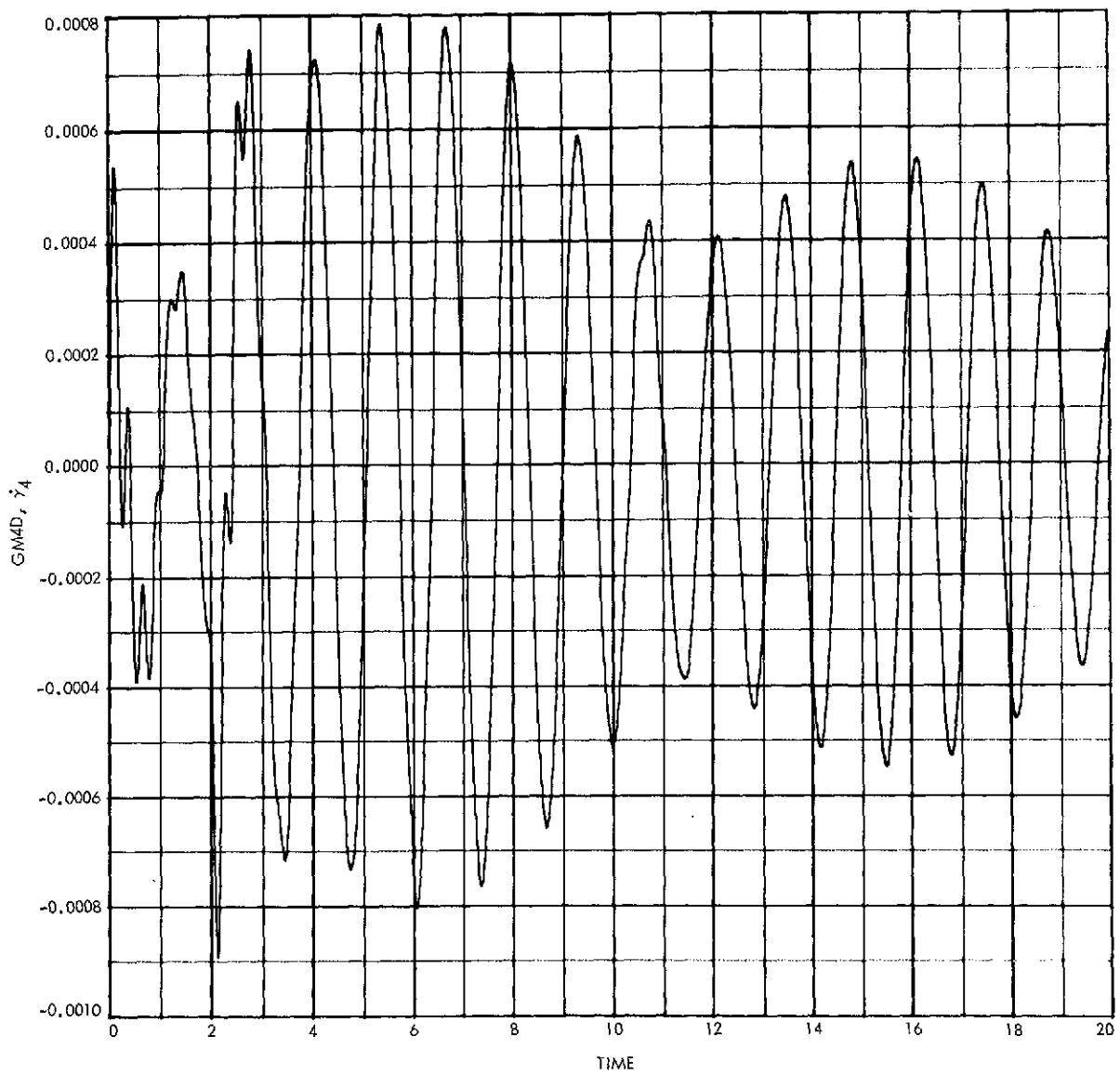


Fig. 16 (contd)

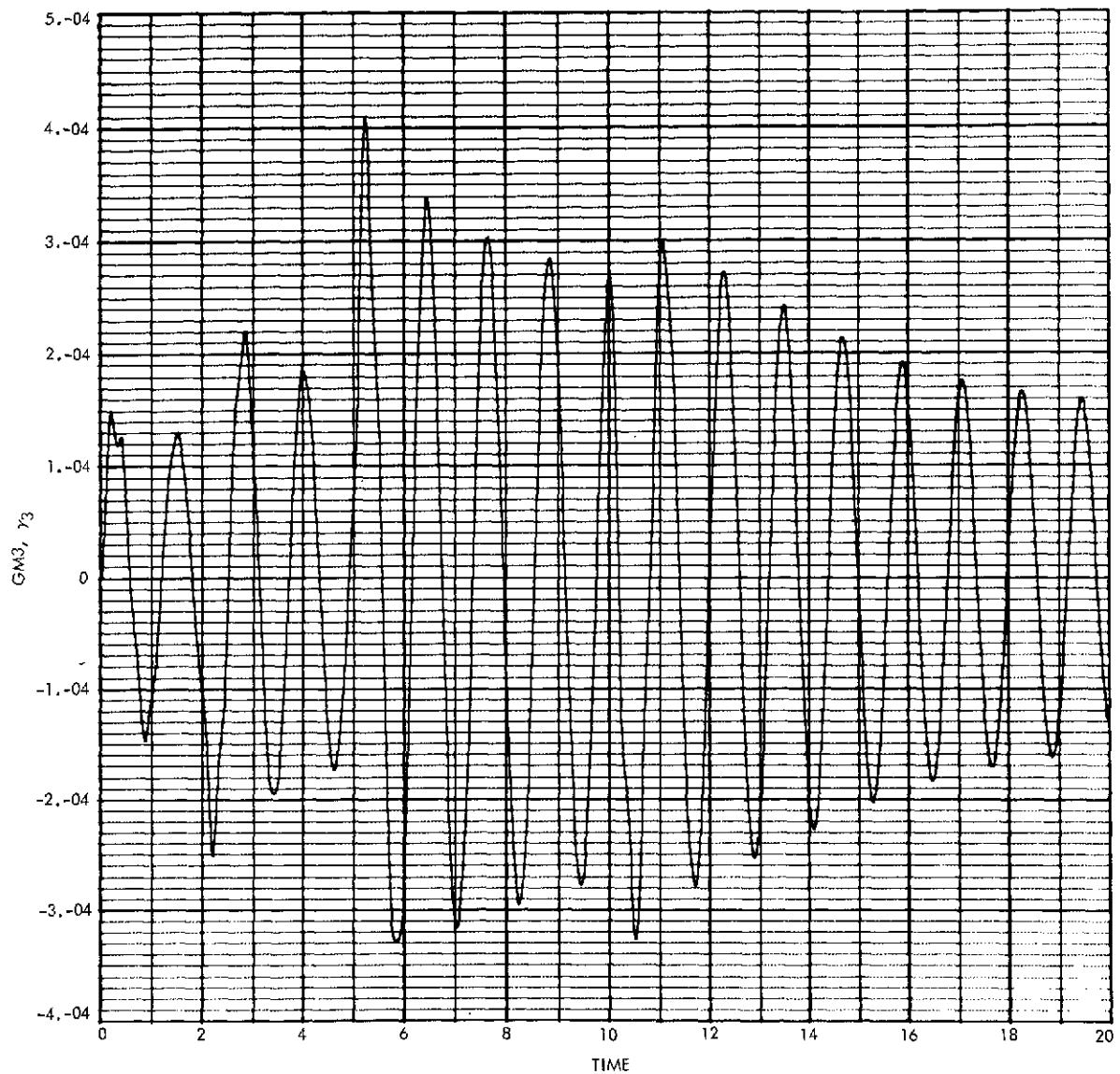


Fig. 16 (contd)

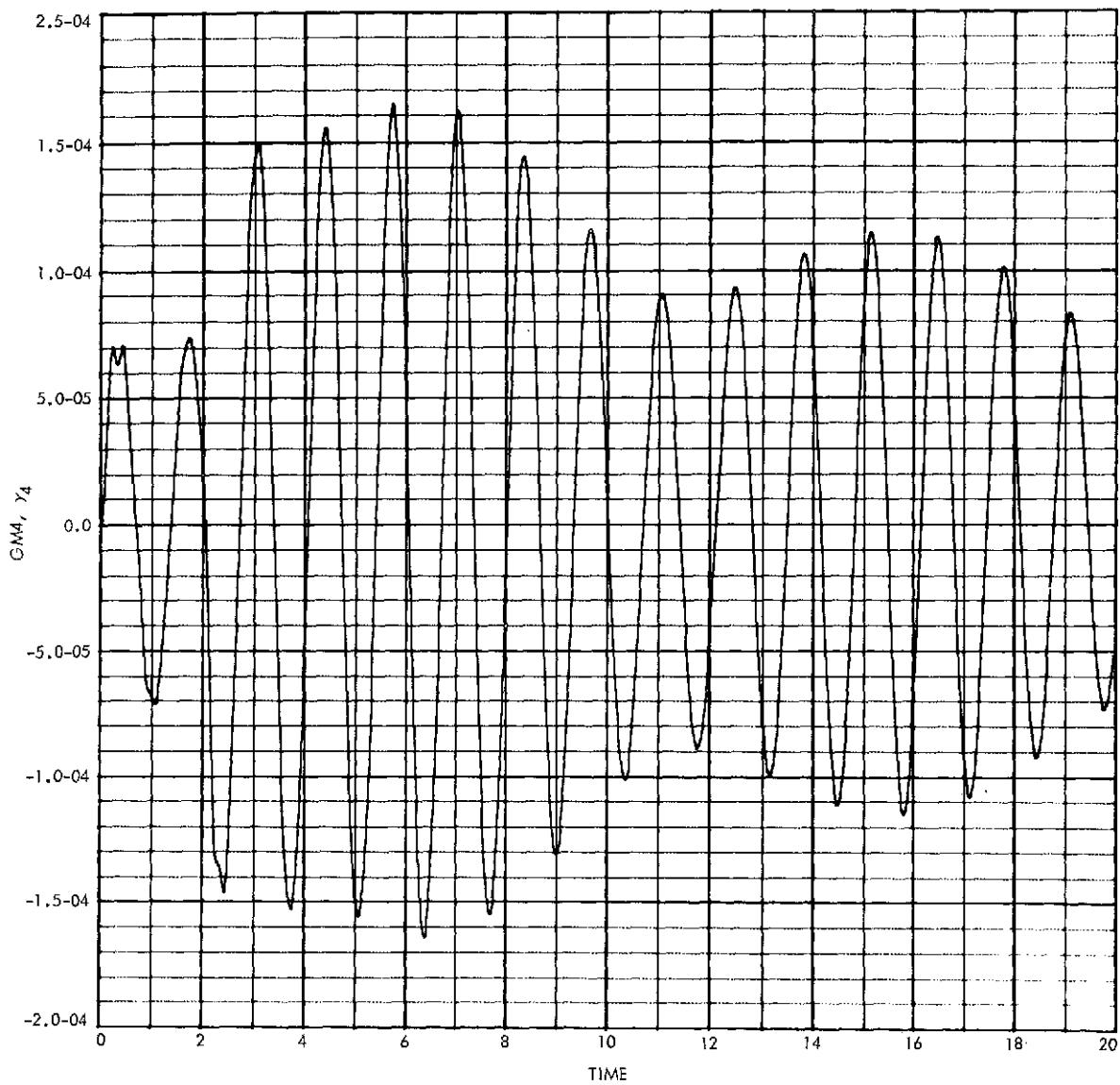


Fig. 16 (contd)

IV. Partially Linearized Systems

A. The Equations

One further distinction that can certainly be made in characterizing systems of hinge-connected rigid bodies (somewhere between those systems where all hinges are permitted *unrestricted* rotation and those systems whose hinge and reference body rotations are *all "small"*) is the partially linearized system. In this report, the term "partially linearized" will refer to the case in which the reference body rotations are unrestricted and some, but not all, of the γ_k ($k \in \mathcal{P}$) and their derivatives are considered to be "small."

The indices of those angles for which linearization is in some sense justified may be collected in the set \mathcal{A} , and, using the expansions of Eq. (9), the linearized form of Eq. (8) becomes

$$(\bar{a}_{00} + \hat{a}_{00}) \dot{\omega}^0 + \sum_{k \in \mathcal{A}} \bar{a}_{0k} \ddot{\gamma}_k + \sum_{k \in \mathcal{P} - \mathcal{A}} (\bar{a}_{0k} + \hat{a}_{0k}) \ddot{\gamma}_k = \sum_{k \in \mathcal{B}} (\bar{E}_k + \hat{E}_k) \quad (15a)$$

$$\begin{aligned} & (\bar{a}_{i0} + \hat{a}_{i0}) \dot{\omega}^0 + \sum_{k \in \mathcal{A}} \bar{a}_{ik} \ddot{\gamma}_k + \sum_{k \in \mathcal{P} - \mathcal{A}} (\bar{a}_{ik} + \hat{a}_{ik}) \ddot{\gamma}_k \\ &= g^{iT} \sum_{k \in \mathcal{B}} \varepsilon_{ik} [\bar{C}^{i0} (\bar{E}_k + \hat{E}_k) + \hat{C}_{i0} \bar{E}_k] + \bar{\tau}_i + \hat{\tau}_i \end{aligned} \quad (15b)$$

It should be noted that, for a wide class of systems, it can be recognized in advance that $\varepsilon_{ik} = 0$ for $i \in \mathcal{A}$ and $k \in \mathcal{P} - \mathcal{A}$. This is the case whenever none of the hinge points whose indices comprise \mathcal{A} lies on a path from ℓ_0 to any hinge point whose index is not in \mathcal{A} . In physical terms, this means that the small-angle rotations are confined to *terminal appendages*, each of which is attached to only one body whose index lies outside of \mathcal{A} . In fact, primary attention is focused on those scalar elements a_{ik} in (15b) where $i, k \in \mathcal{A}$ and i, k are members of the same terminal appendage set. It can be shown that such elements a_{ik} are all constants.

Def. 38. For $v \in \mathcal{B}$, let \mathcal{A}_v denote a "terminal appendage set." Thus \mathcal{A}_v is a subset of \mathcal{A} , containing the body indices of one or more terminal appendages which share a common base body, v .

By appropriate exchanges of rows and columns of the symmetric system matrix, the constant \bar{a}_{ik} 's may be isolated in an $r \times r$ symmetric matrix, A_c , forming the lower right-hand corner of the original matrix. One can then employ a matrix partitioning scheme, where the constant matrix A_c is inverted just once, and a time-varying matrix equation, now of order $(n + 3 - r)$, is solved at every integration step. Clearly, the size of A_c should be maximized for the greatest computational advantage, and, since only one terminal appendage set (there may be several) may be used to form A_c , the largest terminal appendage set should be used.

With this approach, i.e., a single partitioning of the system matrix into constant and time-varying parts, another version of the MBDY subroutine, called MBDYPL, was written to solve the partially linearized case. However, in a move to further simplify and speed up the computational task, Eq. (15) was modified as follows:

$$\hat{a}_{00} \dot{\omega}^0 + \sum_{k \in \mathcal{P}} \hat{a}_{0k} \ddot{\gamma}_k = \sum_{k \in \mathcal{B}} \hat{E}_k \quad (16a)$$

$$\hat{a}_{i0}\dot{\omega}^0 + \sum_{k \in \mathcal{P} - \mathcal{A}} \hat{a}_{ik}\ddot{\gamma}_k + \sum_{k \in \mathcal{A}} \bar{a}_{ik}\ddot{\gamma}_k = g^{i0} \hat{\bar{C}}^{i0} \sum_{k \in \mathcal{P}} \epsilon_{ik} \hat{\bar{E}}_k + \hat{\tau}_i \quad (16b)$$

where

$$\begin{aligned}\hat{\bar{a}}_{00} &= \bar{a}_{00} + \hat{a}_{00} + \dots \\ \hat{\bar{a}}_{0k} &= \bar{a}_{0k} + \hat{a}_{0k} + \dots \\ \hat{\bar{a}}_{i0} &= \bar{a}_{i0} + \hat{a}_{i0} + \dots \\ \hat{\bar{a}}_{ik} &= \bar{a}_{ik} + \hat{a}_{ik} + \dots\end{aligned}$$

The intent of Eq. (16) is to avoid having to form the *separate* quantities \bar{a}_{00} , \hat{a}_{00} , \bar{a}_{0k} , \hat{a}_{0k} , etc., and instead, to compute only the combined forms $\hat{\bar{a}}_{00}$, $\hat{\bar{a}}_{0k}$, etc., using the combined form of the coordinate transformation, $\hat{\bar{C}}^{ik}$. This approach saves overall computation time, although it allows the retention of some second-order terms in the linearized variables that would have disappeared otherwise. Thus, Eq. (16) is the matrix equation actually solved by MBDYPL, a somewhat more accurate representation than that of Eq. (15).

B. Subroutine MBDYPL Usage

Just as in the other subroutines, MBDYPL is entered by means of either of two call statements, one of which initializes the program with the system constants.

Initializing Call Statement

```
CALL MBDYPL(NH,H,MB,MA,PB,PA,G,PI,LI,TI)
```

The call statement arguments are exactly as defined previously, with exception of the two additions

$LI(i)$ = array of linearized variable indicators; $i \in \mathcal{P}$ (If γ_i and its derivatives are assumed small and therefore are to be linearized, $LI(i) = 1$. Otherwise, $LI(i) = 0$.)

$TI(i)$ = array of terminal appendage body indicators; $i \in \mathcal{P}$ (If $TI(i) = 1$, γ_i is a member of the *largest* terminal appendage in the system. If $TI(i) = 0$, it is not, although γ_i could still be a linearized variable.)

It should be made clear that $LI(i)$ is an array to be supplied *by the user*; i.e., it is an *input* array. On the other hand, $TI(i)$ is calculated *by the subroutine* for the user's information, and is an *output* quantity. As mentioned previously, it is desirable to identify the *largest* terminal appendage of the system and its member bodies. An algorithm to do just that is located in the initializing portion of the subroutine, and the results of its calculations are later used for optimum rearrangement and partitioning of the system matrix, as well as made available to the user in TI .

As before, MBDYPL is executed at every numerical integration step using the dynamic call statement.

Dynamic Call Statement

```
CALL MRATE(NH,TH,TB,TA,FB,FA,GM,GMD,GMDD,WO,WDOT)
```

All arguments of the dynamic call statement are exactly as defined for MBDY, with the understanding that $\text{TH}(i)$, $\text{TA}(i,j)$, and $\text{FA}(i,j)$ refer to $\hat{\tau}_i$, \hat{T}^i , and \hat{F}^i , respectively, when $i \in \mathcal{A}$. In general, *any* torque or force expressions should be free of second-order terms in the linearized variables (for which $\text{LI}(i) = 1$) or their derivatives.

The necessary specification statements for MBDYPL are the same as for MBDY, with the addition of the integers LI and TI.

Required Specification Statements

```

INTEGER NH, H(n), PI(n), LI(n), TI(n)

REAL MB(7), MA(n,7), PB(n,3), PA(n,n,3), G(n,3), TH(n),
      TB(3), TA(n,3), FB(3), FA(n,3), GM(n), GMD(n),
      GMDD(n), WO(3)

DOUBLE PRECISION WDOT(n + 3)

```

Again, the statement,

```
PARAMETER Q = n
```

must appear within the subroutine MBDYPL to properly allocate internal storage. A complete listing of MBDYPL is provided in Appendix C.

C. Application of MBDYPL to the Five-Body System

1. Large platform maneuver. Since MBDYPL is designed to accommodate both large and small rotations in a hinge-connected system, Case 1 would be the obvious dynamic maneuver with which to test MBDYPL.

Recall that the maneuver consists of rotating the platform from its initial position ($\gamma_1(0) = 218$ deg, $\gamma_2 = -30$ deg) to the position $\gamma_1 = 168$ deg, $\gamma_2 = 20$ deg by simultaneous slewing about both axes at the commanded rate of 1 deg/s (starting at $t = 0$).

The main simulation program listed in Fig. 6 can be used virtually “as is” by MBDYPL, with only the addition of LI(4) and TI(4) to the INTEGER statement and LI input elements to the DATA statement. In this case, linearization will be requested for γ_3 and γ_4 so that LI(1) = 0, LI(2) = 0, LI(3) = 1, and LI(4) = 1. Examination of TI, after executing “CALL MBDYPL (NH, . . .),” would show it to be identical to LI; i.e., bodies $\mathbf{4}_3$ and $\mathbf{4}_4$ would be identified as members of the largest system terminal appendage. The modified simulation program using MBDYPL is listed in Fig. 17.

Results of the platform maneuver, Case 1, using the partially linearized approach are shown in Fig. 18. Comparison of these to the results produced by MBDY in Fig. 7 shows the differences to be indistinguishable. Figure 19 expands the plots of ω^o components for just the first 10 s for closer comparison (with Fig. 8).

While the accuracy of the MBDYPL run in this case was apparently excellent, the speed advantage over MBDY was only very slight, i.e., 430 versus 440 s of computation time for this four-hinge system. Since MBDYPL does perform a number of manipulations associated with partitioning and rearranging the system matrix as well as matrix multiplication to solve the partitioned system, some tradeoff between

the time *used* to accomplish these tasks and the time *saved* by recognizing certain constants and advanced matrix inversion should be expected. Just exactly where the "break-even" point between MBDY and MBDYPL speed is depends on both the number of hinges and the number of linearized hinges. However, in the case simulated here, we seem to be quite near the break-even point. In contrast, the system simulation in Ref. 7 with 14 hinges (10 linearized) was speeded up by a factor of two using MBDYPL.

It should also be noted that the subroutine MBDYPL does accommodate *prescribed* motions at *any* of its hinges, as do the other subroutines, whether the hinge is linearized or not.

V. Summary and Conclusions

This report confines itself to the consideration of mathematical models useful for the description of hinge-connected rigid body system dynamics and, further, the computer simulation of those idealizations using general-purpose subroutines. As the sequel to Ref. 9, it documents in explicit detail the application of each of these subroutines under the circumstances for which they were especially designed, i.e., MBDY for unrestricted systems, MBDYTL and MBDYL for totally linearized systems, and MBDYPL for partially linearized systems. In each case, the subroutines, which are written in Fortran V, serve as the powerful tool that is called upon by the main simulation program (written here in CSSL III) to calculate vehicle attitude dynamics.

In the process of drawing conclusions about the relative utility of these programs, one must certainly begin with the obvious conclusion that MBDY, since it solves the exact and unrestricted equations of motion for the system, serves as the ultimate standard of accuracy for the rest. However, MBDY need not be restricted in its use to just the role of a "check routine." For small systems, it will run faster than MBDYPL, and one can certainly envision system rotations large enough to make the linearized routines completely unsuitable for the simulation accuracies required. Further, the judicious use of the option available in all the routines for *prescribed* hinge rotations can speed the running time of a simulation using MBDY remarkably. Thus, MBDY should be used often and to good advantage.

Of course, the utility of the completely linearized dynamic simulation subroutines, MBDYTL and MBDYL, was well exposed in Section III. For large or small systems of bodies in which all the hinge and reference body rotations are deemed "small," either due to fortunate circumstance or deliberately controlled constraint, the use of MBDYTL or MBDYL can offer tremendous speed advantages over MBDY and still maintain acceptable accuracies of the dynamic response. This totally linearized approach has proved to be of the greatest value in analyzing control problems at JPL, since most systems fall, or can be forced into, this category. In fact, the accuracy of MBDYL has been more than adequate for a large variety of space vehicle control problems studied at JPL. Of course, if the vehicle *control* dynamics can also be linearized, the inverted attitude dynamics matrix used by MBDYTL or MBDYL can be used directly in control stability analyses, i.e., to compute overall system root loci and/or eigenvalues.

Finally, the utility of MBDYPL is perhaps the least of the four routines, although it still offers some advantages, particularly for large systems. Where there are a

few, very large hinge rotations to be simulated but the large majority of system hinges exhibit only small excursions, MBDYPL may offer the only chance to achieve the desired accuracies in a reasonable simulation time. In similar circumstances, it may serve as a more *economical* reference or check solution source than MBDY. All in all, the partitioning scheme employed in MBDYPL, while not offering the tremendous speed advantages of MBDYL, at least provides some gain in efficiency in certain specialized instances.

As for the general-purpose character of all four subroutines, it is felt that their presentation in Fortran will make them widely useful, both to those who are comfortable with special-purpose simulation languages and to those who prefer to write their own simulation programs and integration subroutines. In order to make the routines more easily accessible, the entire package has been submitted to COSMIC (Computer Software Management and Information Center), University of Georgia, Athens, GA 30601, for evaluation and dissemination to interested governmental agencies as well as industrial, educational, and research institutions. While many may wish to further specialize one or more of these programs to their own particular needs, the four programs, as they are described here, can serve as a reliable and valuable addition to any simulation subroutine library.

```

PROGRAM 5-BODY SYSTEM SIMULATION USING MBDYPL --- CASE 1
  INTEGER NH,PI(4),H(4),LI(4),TI(4)
  ARRAY MB(7),MA(4,7),PB(4,3),PA(4,4+3),G(4+3),TB(3),TA(4,3)
  ARRAY FB(3),FA(4,3),TH(4),GM(4),GMD(4),GMDD(4),W0(3)
  DOUBLE PRECISION W0DT(7)
  DATA H/0.,1.,0.,3./PI/0.,0.,0./LI/0.,0.,1.,1/
  DATA MB/115.,315.,440.,+14.,-14.,34.,6.+410./
  DATA MA(1,1)/-.35/MA(1,2)/.35/MA(1,7)/6.8/
  DATA MA(2,1)/4.85/MA(2,2)/2.2/MA(2,3)/5.5/MA(2,4)/-.41/MA(2,5)/.07/
  DATA MA(2,6)/-.54/MA(2,7)/57.5/
  DATA MA(4,1)/27.2/MA(4,2)/.2/MA(4,3)/27.2/MA(4,7)/10.7/
  DATA FB(1,3)/-1.5/PB(3,2)/-1.2/
  DATA PA(1,1+3)/-.75/PA(1,2+2)/-.10/PA(1,2+3)/-.75/
  DATA FA(2+2,2)/-.22/PA(2+2,3)/.20/
  DATA PA(4,4+2)/3.3/
  DATA G(1,3)/1./G(2,1)/-1./G(3,1)/1./G(4,3)/1./
  CONSTANT TM=.020,DE=.005,TR=2.0
  CONSTANT TFINAL=60.,CLKTIM=1100.
  CONSTANT L1=.23,L2=.21,L3=.31
  CONSTANT SLRAT=.01745329
  CONSTANT PHIZ=0.,THETZ=0.,PSIZE=0.
  CONSTANT W1I=0.,W2I=0.,W3I=0.
  CONSTANT K1=3500.,K2=3500.,K3=2000.,K4=2000.
  CONSTANT B1=20.,B2=20.,B3=10.,B4=10.
  CONSTANT IC1=218.,IC2=-30.

INITIAL
  NH=4
  RD=57.2957795 $ DR=1./RD $ GM1CI=IC1/RD $ GM2CI=IC2/RD
  GM1I=IC1/RD $ GM2I=IC2/RD
  CALL MBDYPL(NH,H,MB,MA,PB,PA,G,PI,LI,TI,
  PZI,PAI,PBI,PCI=HCK(INITL,PHIZ,THETZ,PSIZE))

END
DYNAMIC
  IF(TIME.GE.TFINAL) GO TO S1
DERIVATIVE BODY5
  VARIABLE TIME=0.
  CINTERVAL CT=.06
  XERROR W1=1.E-6,W2=1.E-6,W3=1.E-6,GM3=1.E-4
  STPCLK CLKTIME
  OUTPUT 50.,W1D,W2D,W3D,W1,W2,W3,THET1,THET2,THET3+...
  IN1,IN2,IN3,GM1,GM2,GM3,GM4,PL1,PL2
  PREPAR W1,W2,W3,THET1,THET2,THET3,GM1D,GM2D+...
  GM1,GM2,GM3D,GM4C,GM3,GM4
NOOPT
  W0(1)=W1 $ W0(2)=W2 $ W0(3)=W3
  PL1=RD*GM1 $ PL2=RD*GM2
  GM(1)=GM1 $ GM(2)=GM2 $ GM(3)=GM3 $ GM(4)=GM4
  GMD(1)=GM1D $ GMD(2)=GM2D $ GMD(3)=GM3D $ GMD(4)=GM4D
COMMENT
COMMENT CELESTIAL-SENSOR-MEASURED ANGLES
COMMENT
  D1,D2=HCK(MATRIX,PZ0,PA0,PB0,PC0)
  NX0,NY0,NZ0=HCK(ITOB,0.,0.,1.,D1,D2)
  LX0,LY0,LZ0=HCK(ITOB+1.,0.,0.,1.,D1,D2)
  THET1=ATAN2(NY0,NZ0) $ THET2=ATAN2(-NX0,NZ0)
  THET3=ATAN2(-LY0,LX0)

```

Fig. 17. Case 1 simulation program using MBDYPL

```

COMMENT
COMMENT      RATE FEEDBACK SIGNAL AND SWITCHING AMP. DYNAMICS
COMMENT
IN1=-THET1-TR*W1 $ IN2=-THET2-TR*W2 $ IN3=-THET3-TR*W3
AMP1=SWAMP(TIME,TH,DB,IN1)
AMP2=SWAMP(TIME,TH,DB,IN2)
AMP3=SWAMP(TIME,TH,DB,IN3)

COMMENT
COMMENT      PLATFORM POSITION COMMANDS
COMMENT
GM1CD=SLRAT*(-1.+STEP(50.0,TIME)) $ GM1C=INTEG(GM1CD,GM1CI)
GM2CD=SLRAT*(1.-STEP(50.,TIME)) $ GM2C=INTEG(GM2CD,GM2CI)

COMMENT
COMMENT      REFERENCE BODY EXTERNAL TORQUE
COMMENT      HINGE TORQUES
COMMENT
TB{1}=L1*AMP1 $ TB{2}=L2*AMP2 $ TB{3}=L3*AMP3
TH{1}=-K1*(GM1-GM1C)-B1*GM1D
TH{2}=-K2*(GM2-GM2C)-B2*GM2D
TH{3}=-K3*GM3-B3*GM3D
TH{4}=-K4*GM4-B4*GM4D

COMMENT
COMMENT      SYSTEM ANGULAR ACCELERATIONS
COMMENT
CALL MRATE(NH,TH,TB,TA,FB,FA,GM,GMD,GMDD,W0,WDOT)
W1D=WDOT(1) $ W2D=WDOT(2) $ W3D=WDOT(3)

COMMENT
COMMENT      SYSTEM ANGULAR RATES AND POSITIONS
COMMENT
W1=INTEG(WDOT(1),W1I)
W2=INTEG(WDOT(2),W2I)
W3=INTEG(WDOT(3),W3I)
GM1D=INTEG(WDOT(4),0.) $ GM2D=INTEG(WDOT(5),0.)
GM3D=INTEG(WDOT(6),0.) $ GM4D=INTEG(WDOT(7),0.)
GM1=INTEG(GM1D,GM1I) $ GM2=INTEG(GM2D,GM2I)
GM3=INTEG(GM3D,0.) $ GM4=INTEG(GM4D,0.)

COMMENT
COMMENT      HCK PARAMETER RATES AND POSITIONS
COMMENT
PZOD+PAOD+PBOD,PCOD=HCK(HCK,PZ0+PA0+PB0,PC0,W1,W2,W3)
PZ0=INTEG(PZOD,PZI)
PA0=INTEG(PAOD,PAI)
PB0=INTEG(PBOD,PBI)
PC0=INTEG(PCOD,PCI)

END
END
END
TERMINAL
S1.. CONTINUE
END
END

```

Fig. 17 (contd)

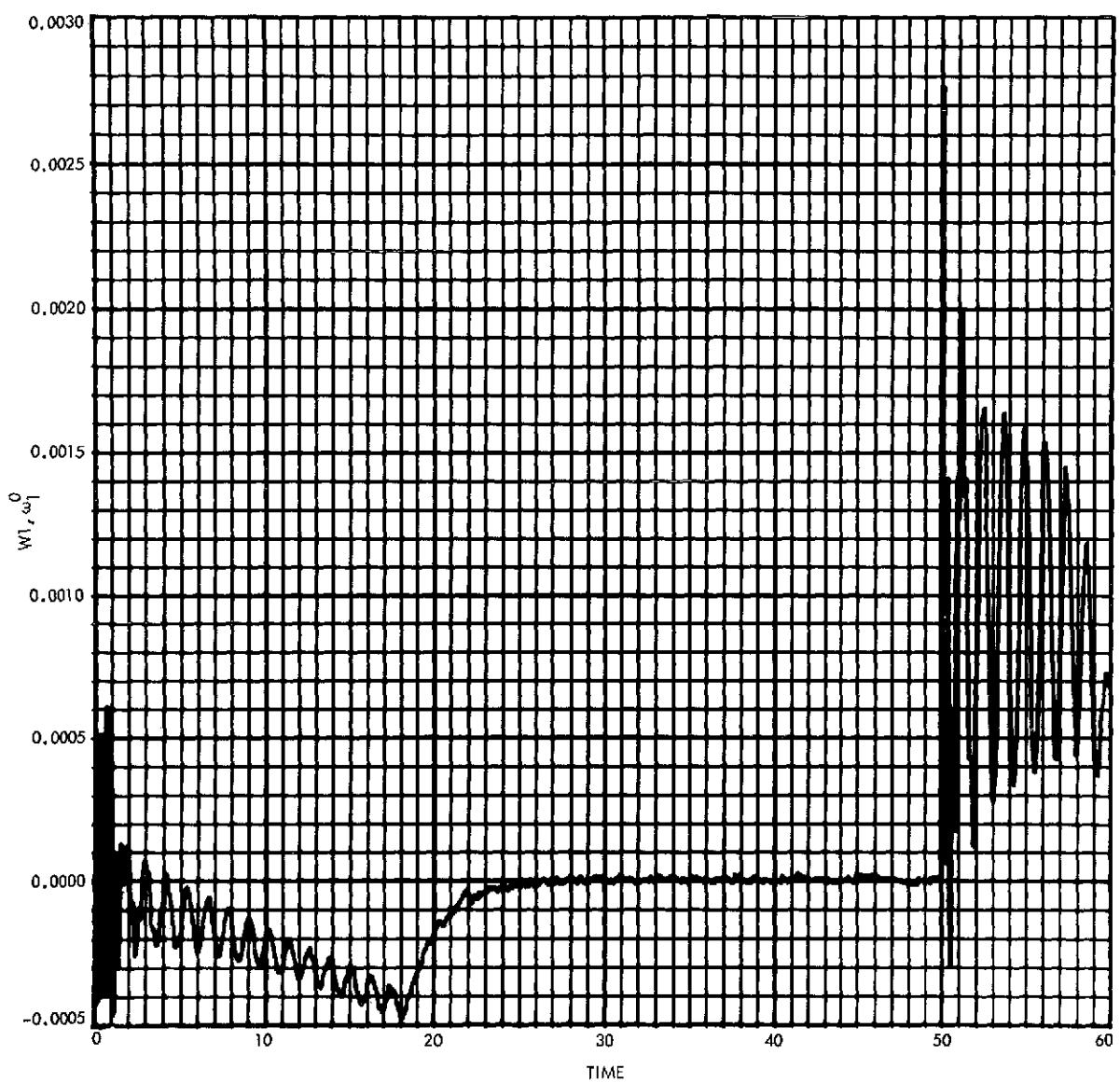


Fig. 18. Results of Case 1 simulation using MBDYPL

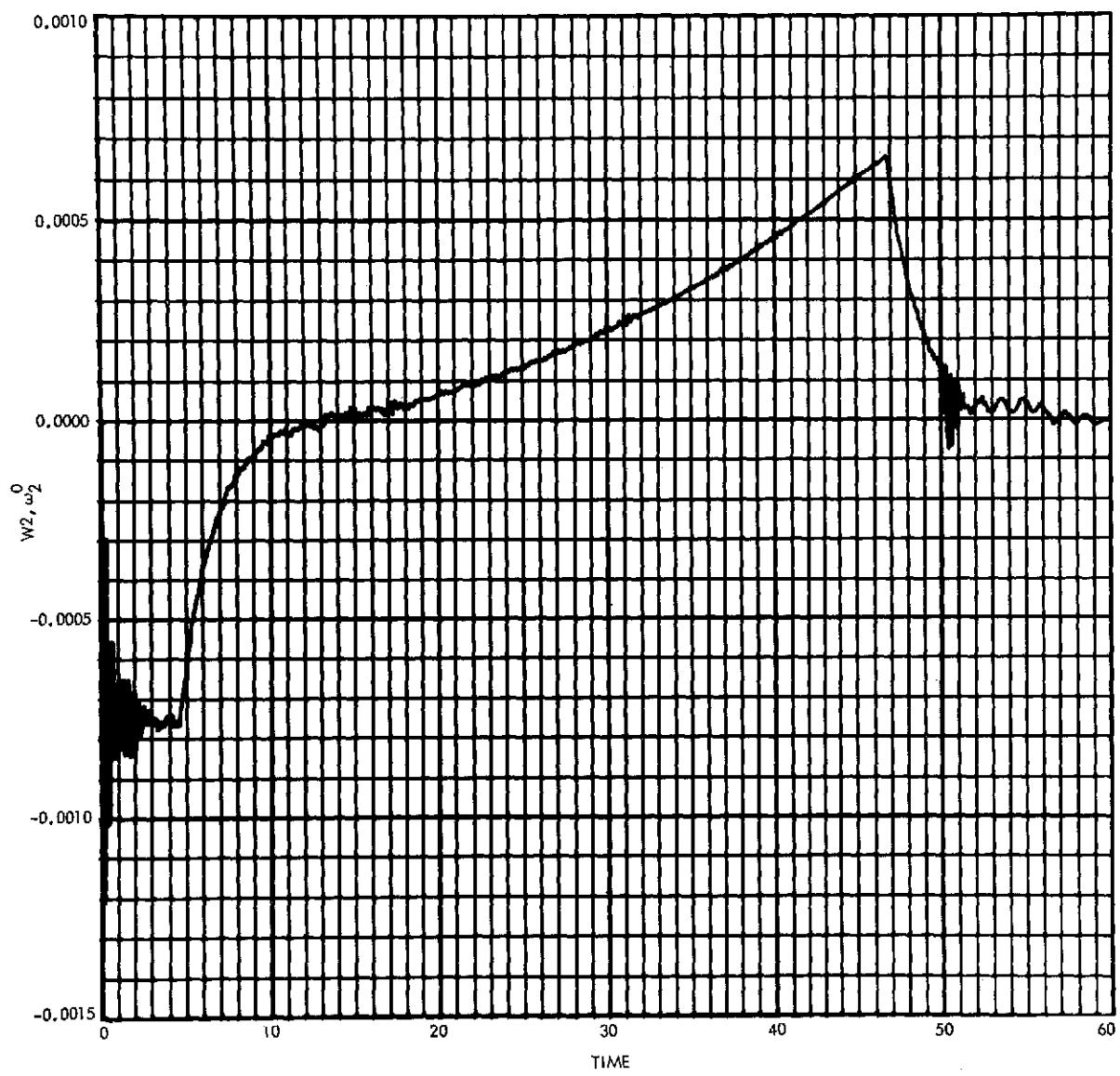


Fig. 18 (contd)

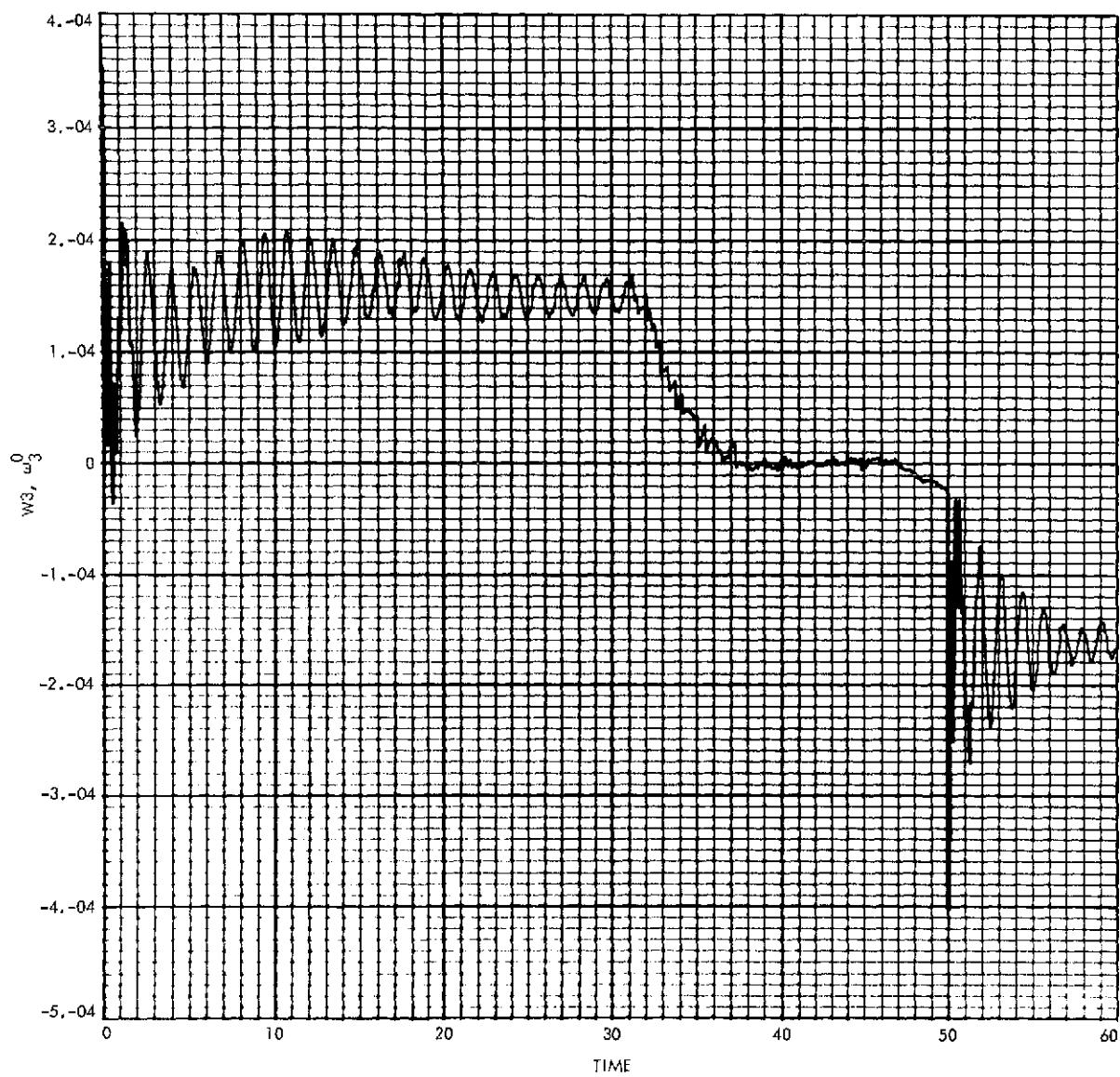


Fig. 18 (contd)

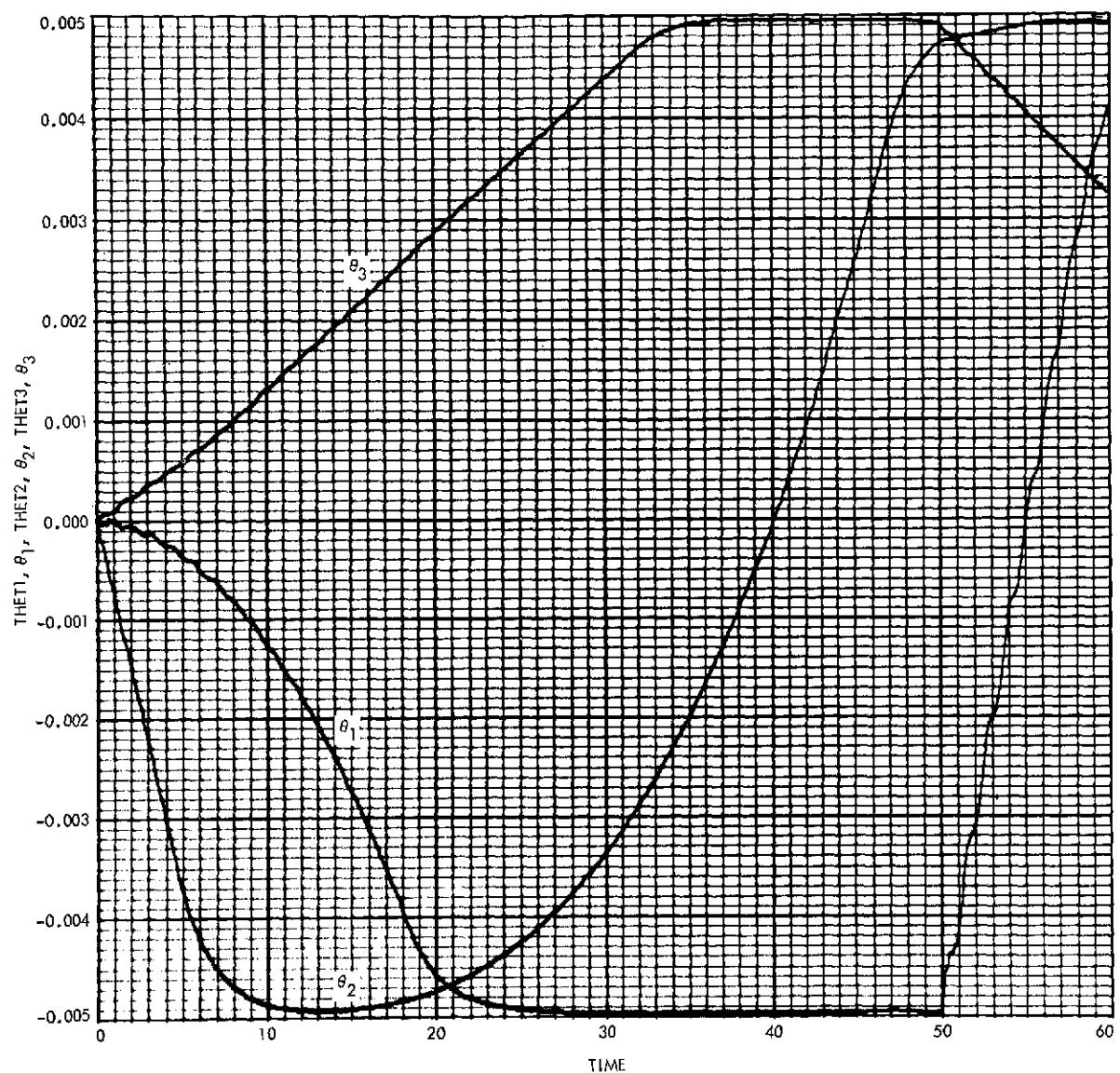


Fig. 18 (condt)

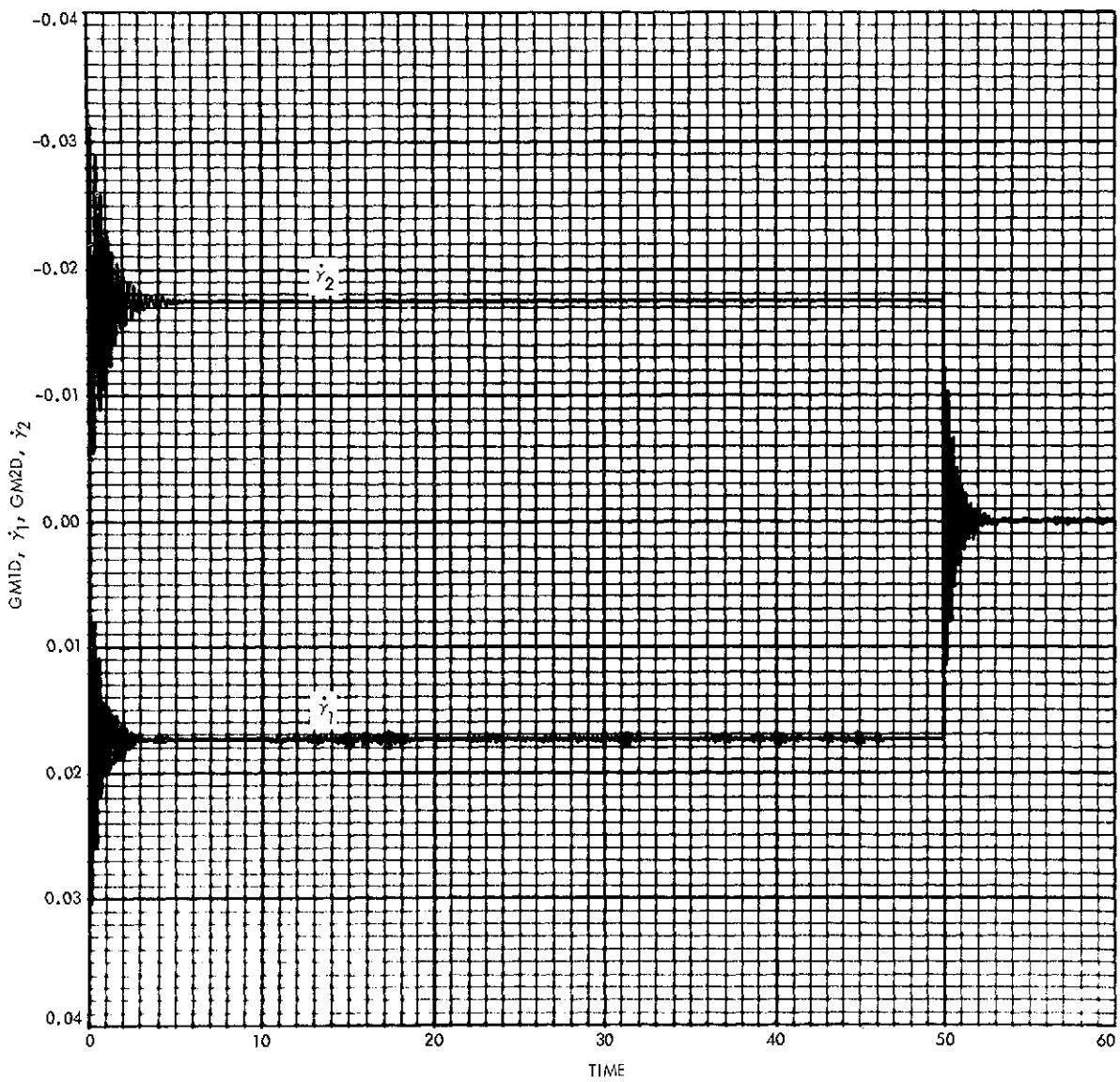


Fig. 18 (contd)

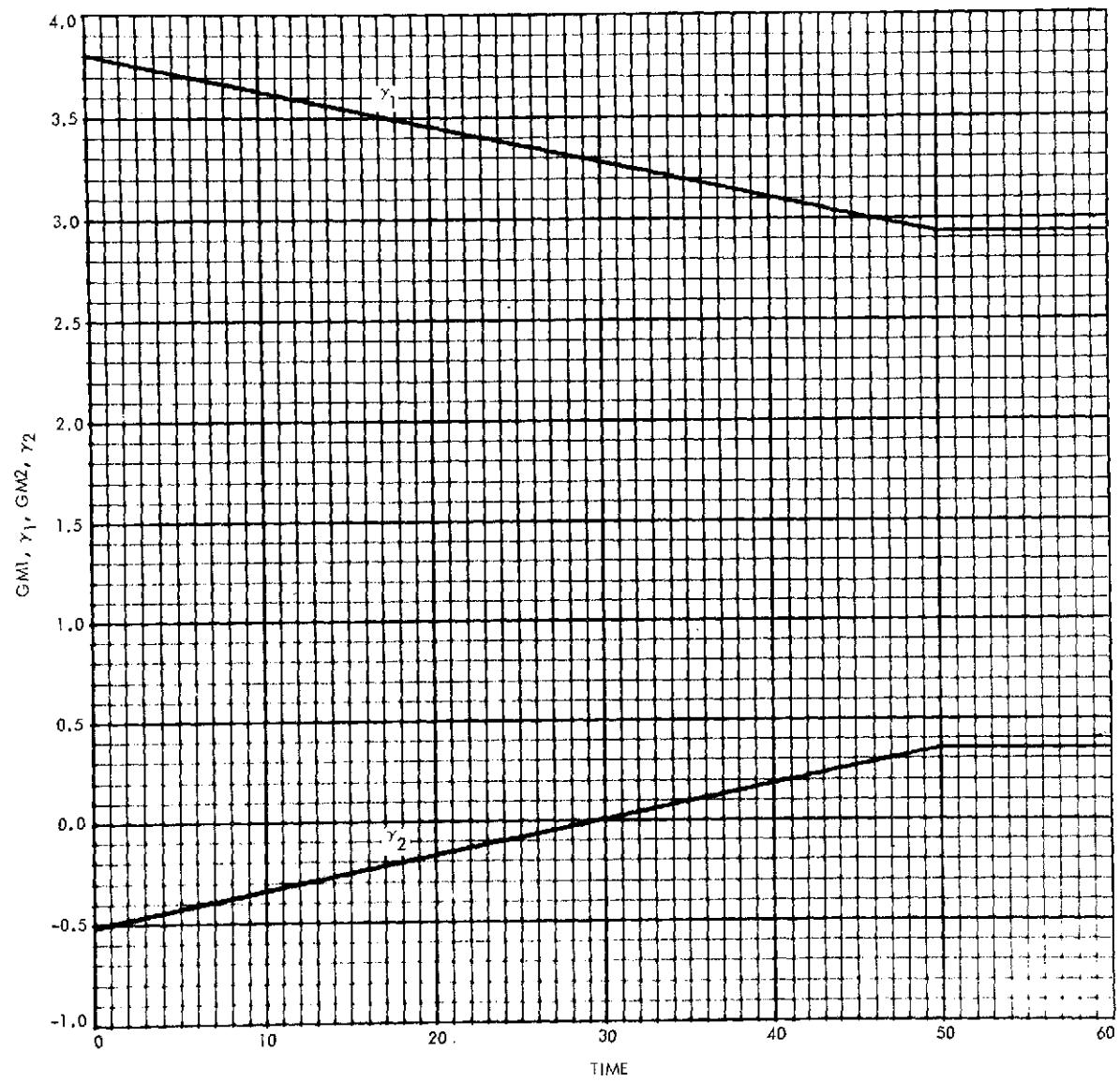


Fig. 18 (contd)

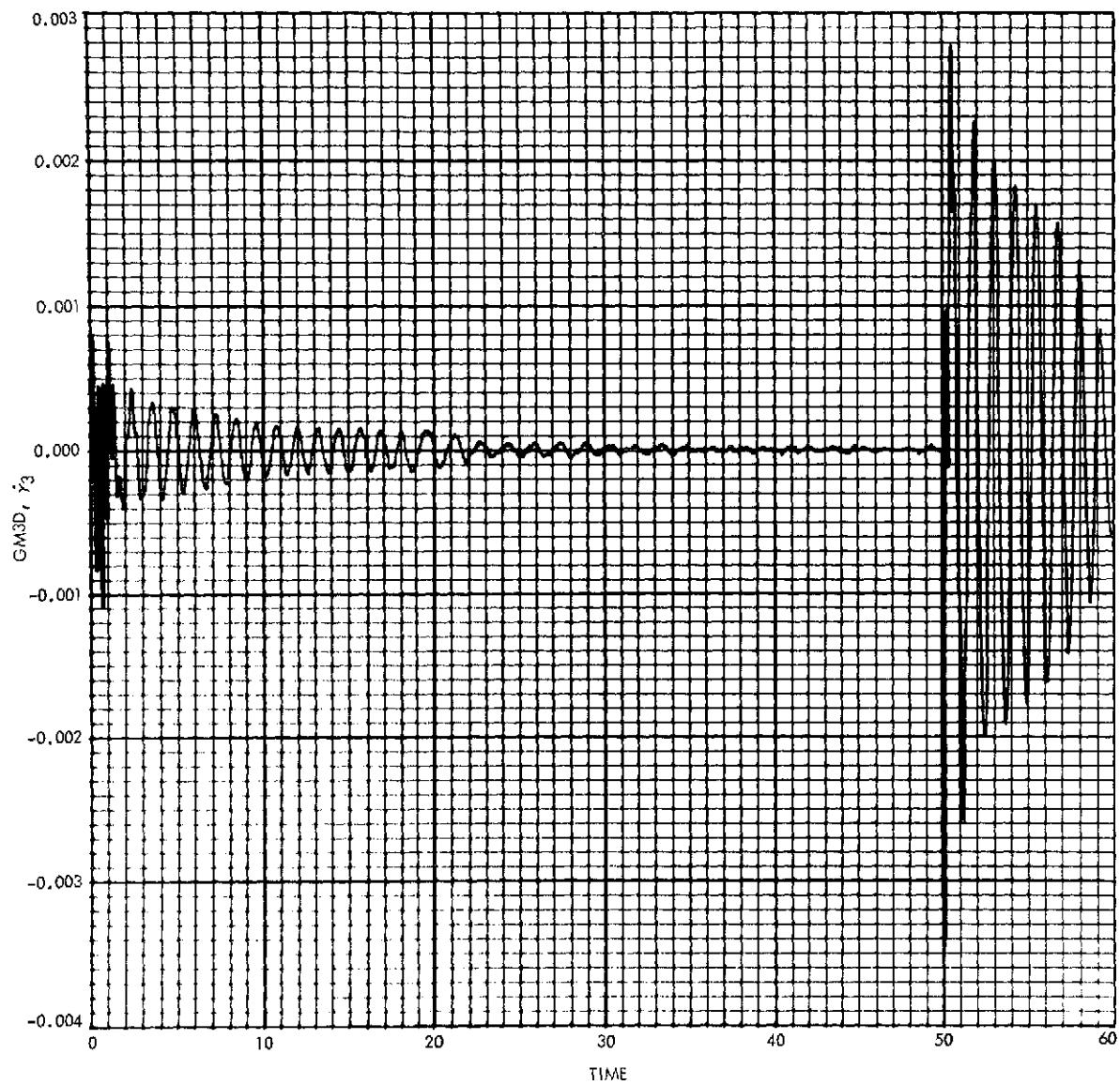


Fig. 18 (contd)

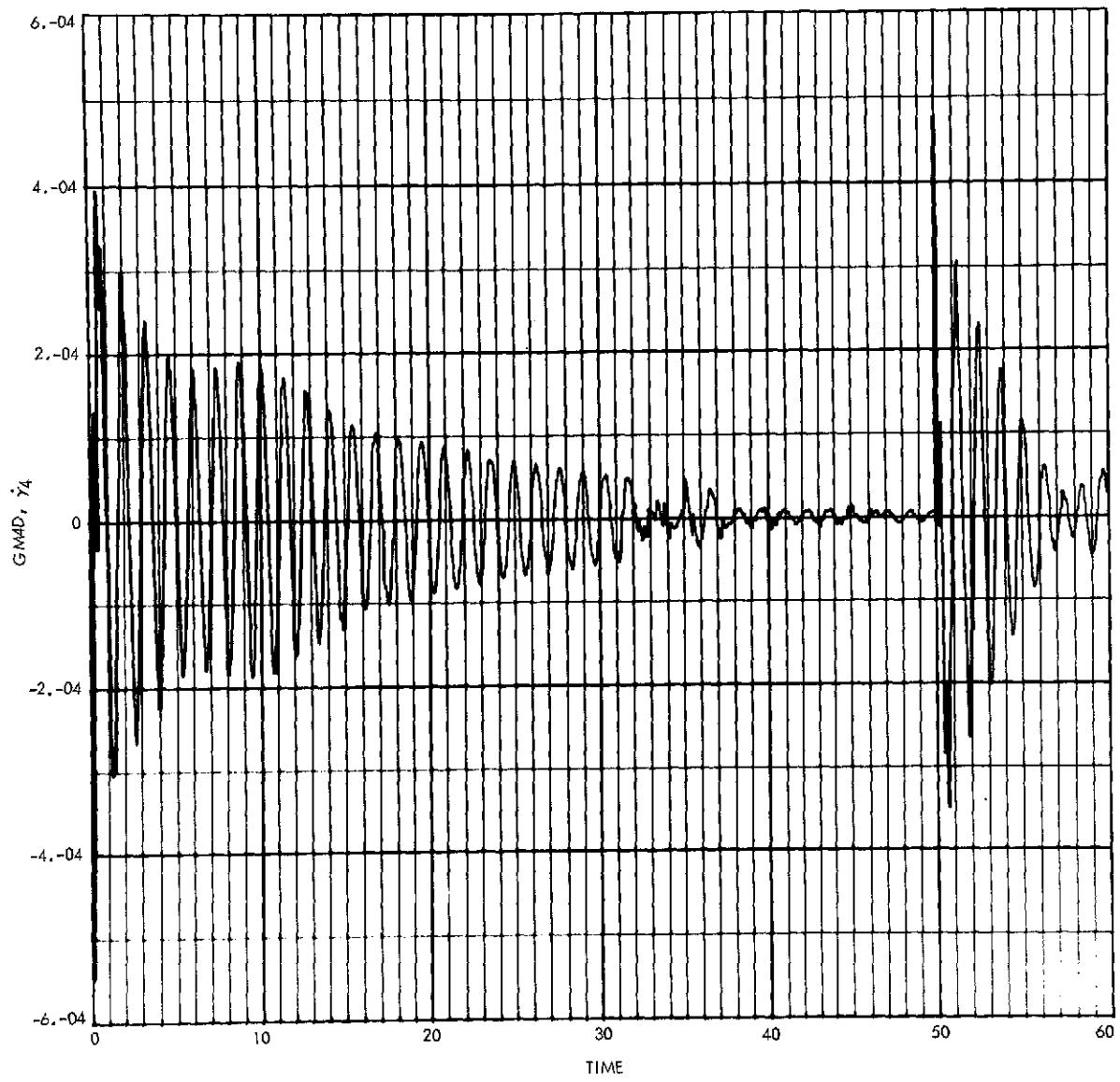


Fig. 18 (contd)

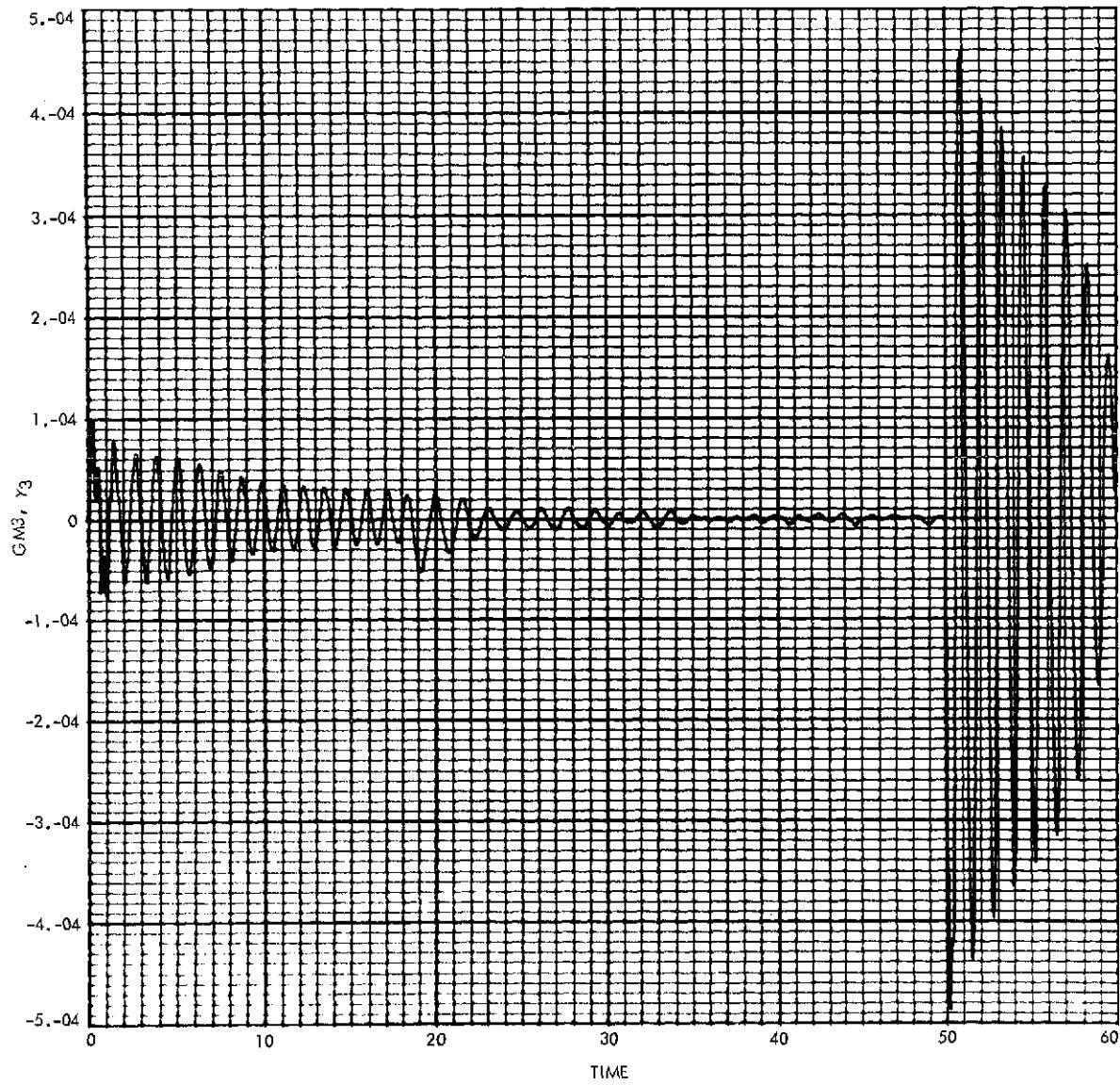


Fig. 18 (contd)

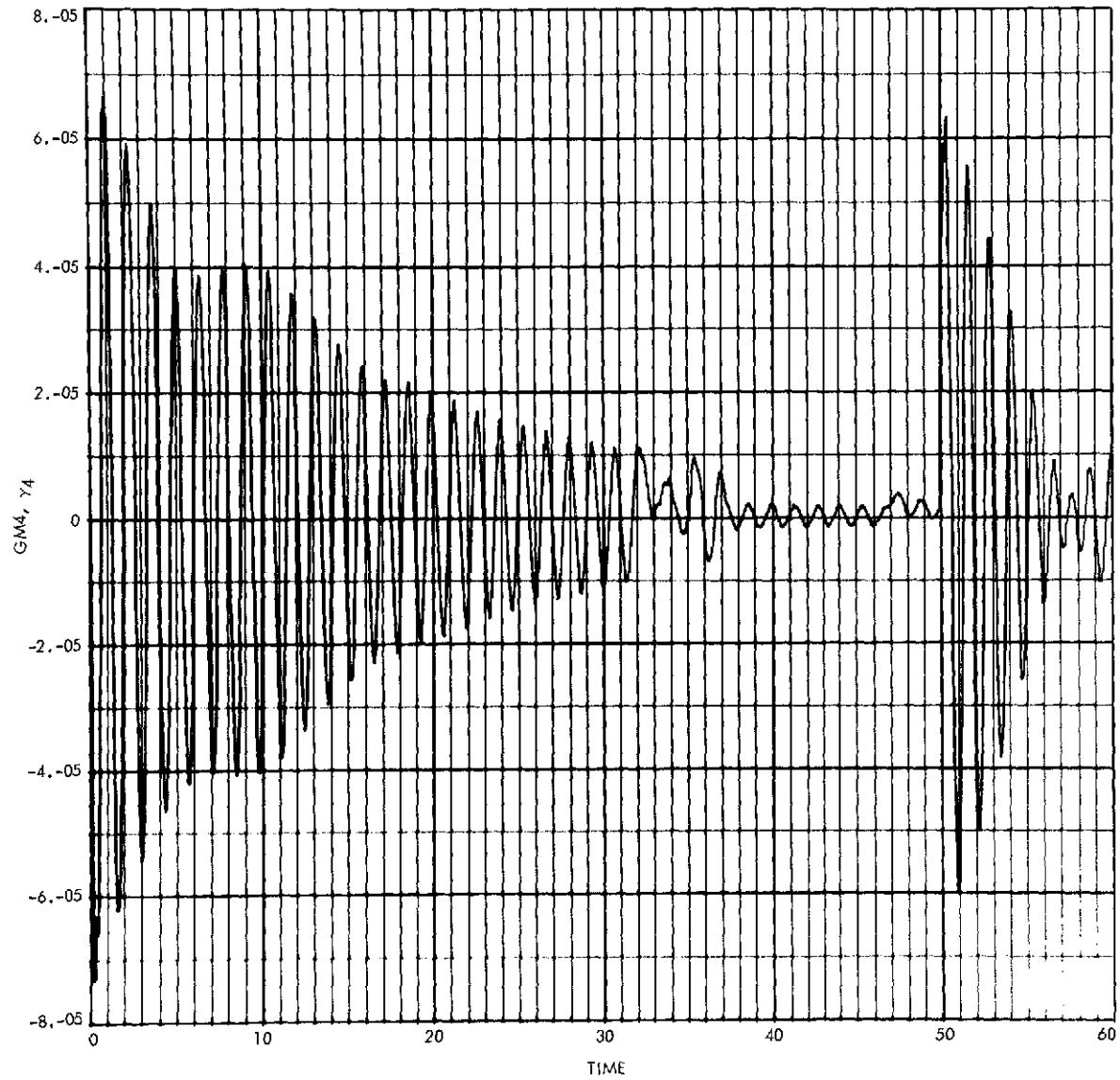


Fig. 18 (contd)

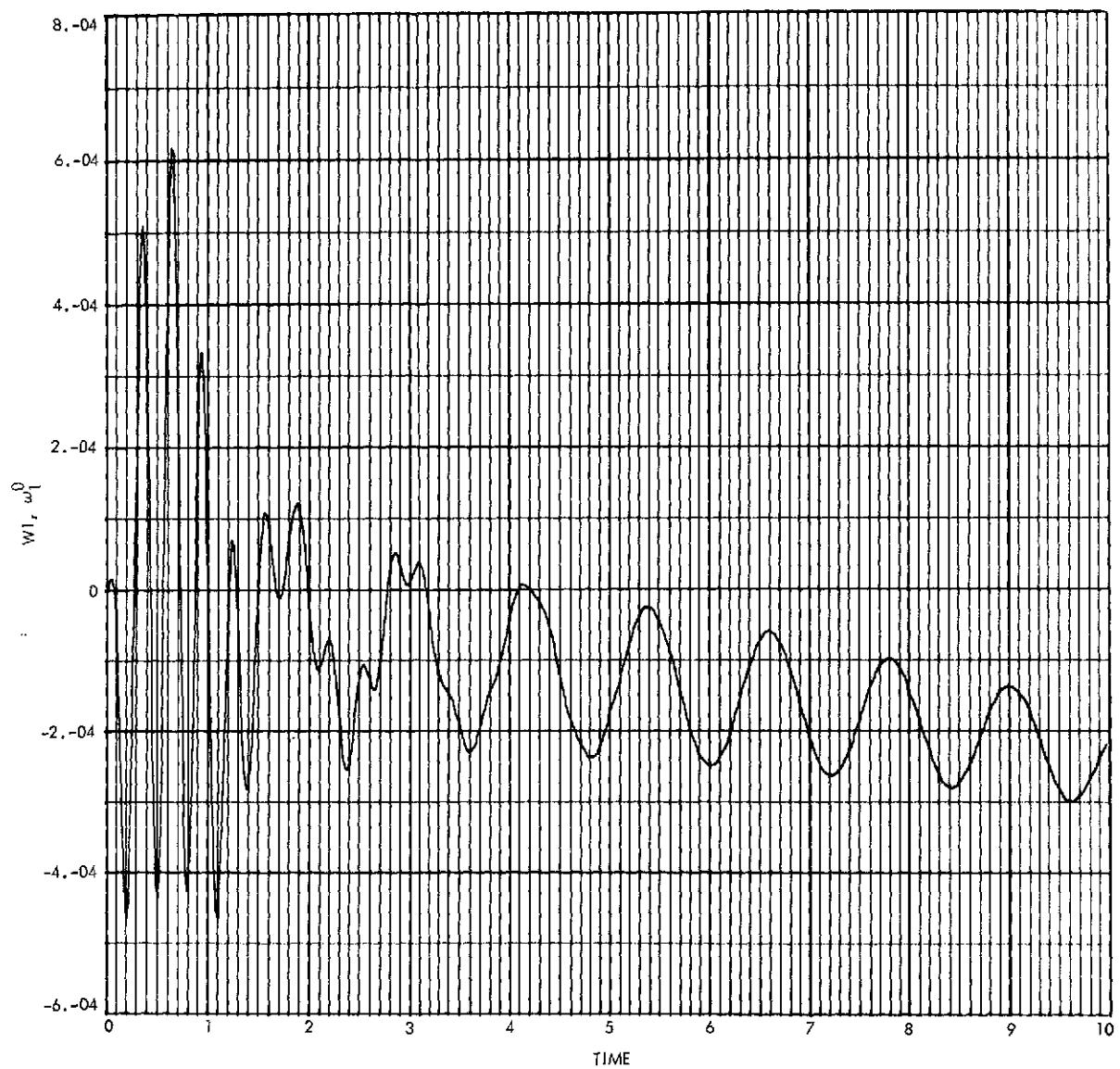


Fig. 19. Results of Case 1 simulation using MBDYPL (components of ω ; first 10 s)

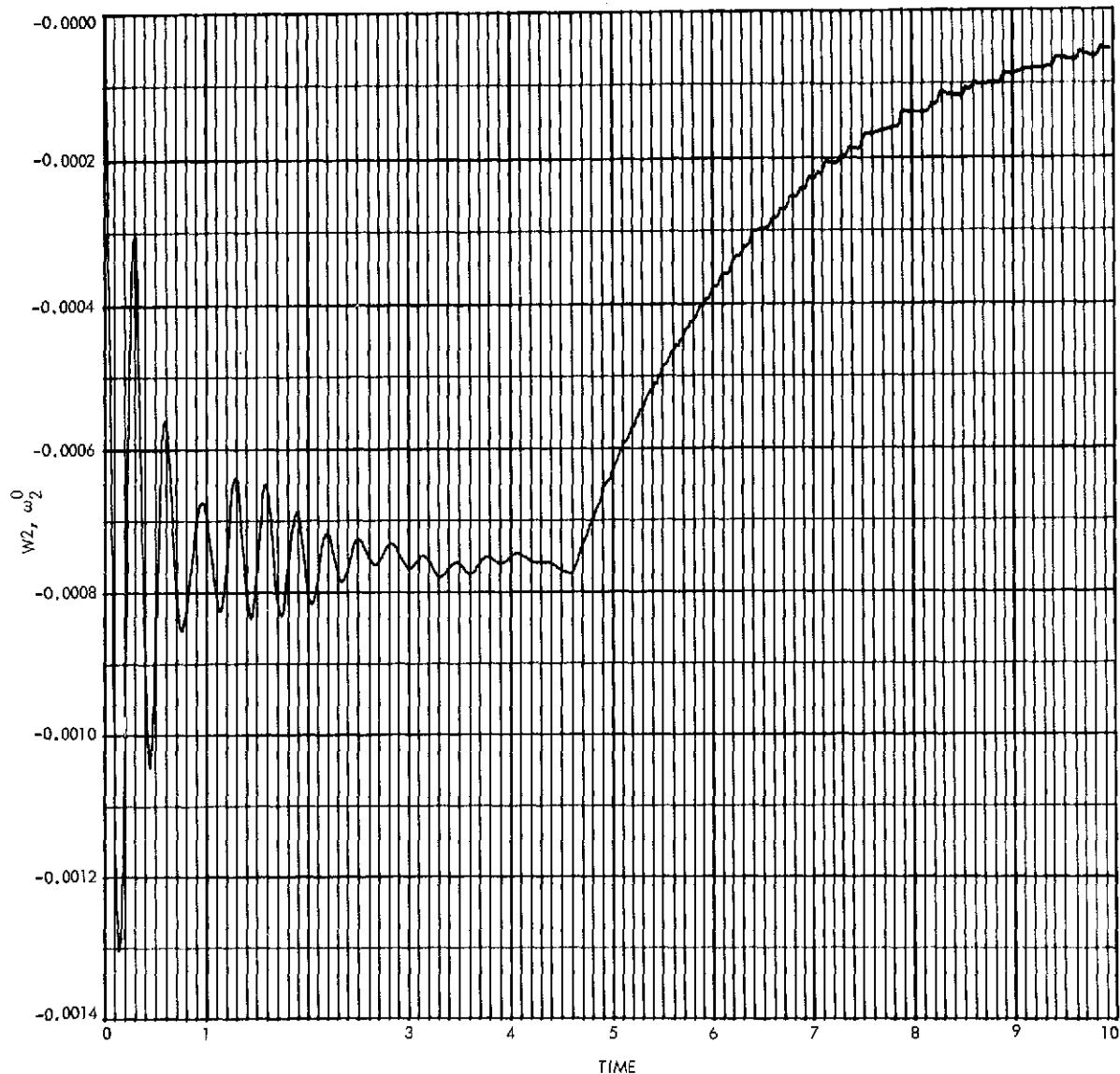


Fig. 19 (contd)

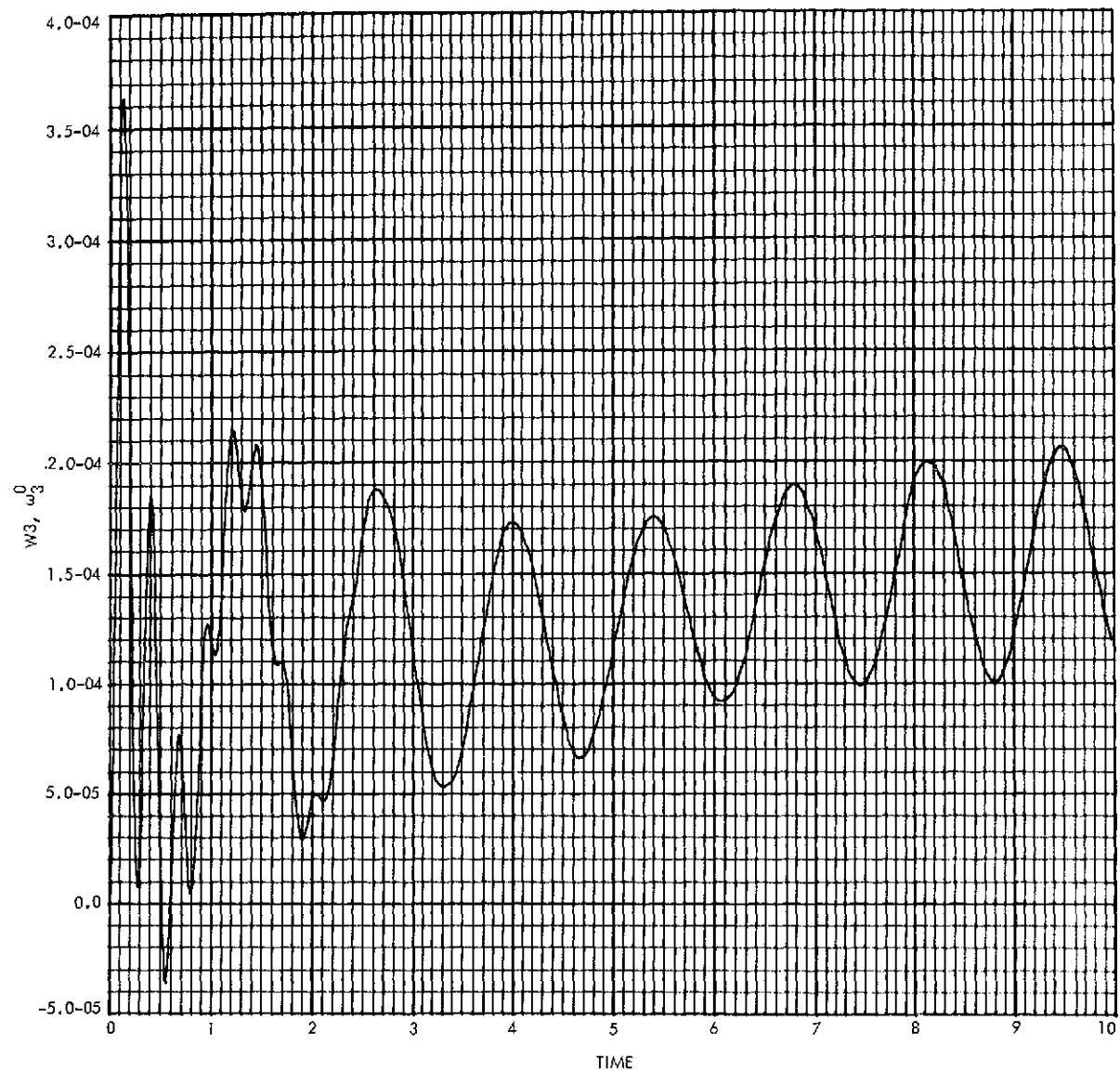


Fig. 19 (contd)

Appendix A

Subroutine MBDY Description

I. Algorithm for Calculating Direction Cosine Matrices, C^{0r} ($r \in \mathcal{P}$)

As was pointed out in Section IIC, all the coordinate transformation, or direction cosine, matrices, C^{rj} ($r, j \in \mathcal{B}$) need not be calculated if all the system equations are expressed in the reference body (\mathcal{A}_0) coordinate frame. Thus, the objective is simply to obtain C^{0r} for all $r \in \mathcal{P}$. An algorithm for accomplishing this is incorporated in MBDY and in MBDYTL and MBDYPL as well (MBDYL uses no direction cosine transformations). The computation procedure is as follows:

Step 1. Let $r = 1$

Step 2. Calculate elements of the direction cosine matrix C^{ir} , where $i = h_r$ (see Eq. 6)

Step 3. If $j = 0$, jump to Step 5

Step 4. Calculate elements of C^{0r} from relation $C^{0r} = C^{0j}C^{jr}$

Step 5. Let $r = r + 1$

Step 6. If $r < n + 1$, go back to Step 1. Otherwise, stop.

The calculations involved in Step 2, of course, may include the use of $\sin \gamma_r$ and $\cos \gamma_r$, or they may use approximations thereof (i.e., $\gamma_r = \sin \gamma_r, \cos \gamma_r = 1$), depending upon whether linearization is being employed or not. This sort of bootstrapping calculation of C^{0r} , by way of Step 4, is made possible by the fact that j is always less than r , and therefore C^{0j} and C^{jr} must necessarily have been computed before coming to C^{0r} .

II. Summary of MBDY User Requirements

Subroutine Entry Statements

```
CALL MBDY(NH,H,MB,MA,PB,PA,G,PI)
```

```
CALL MRATE(NH,TH,TB,TA,FB,FA,GM,GMD,GMDD,WO,WDOT)
```

Input/Output Variable Type and Storage Specifications—Fortran V

INTEGER NH, H(*n*), PI(*n*)

REAL MB(7), MA(*n*,7), PB(*n*,3), PA(*n*,*n*,3), G(*n*,3), TH(*n*),
TB(3), TA(*n*,3), FB(3), FA(*n*,3), GM(*n*), GMD(*n*),
GMDD(*n*), WO(3)

DOUBLE PRECISION WDOT(*n* + 3) (where *n* = number of system hinges)

External Subroutines Called

SOD—double precision matrix equation solving subroutine; user-supplied (see line 437 in MBDY listing)

This subroutine solves the matrix equation $A\ddot{x} = EC$, where A is of order $IV \times IV$ and $IV \geq 3$. The solution is returned in EC .

Subroutine Setup

Insert the Fortran V statement (see listing)

PARAMETER Q = n

where n = number of system hinges.

Data Restrictions

$n \geq 1$

Core Storage Required

Code: 2000 words

Data: 500 words (minimum; increases with n)

Listing

```

1*      SUBROUTINE MBODY(NH,H,MA,MB,PA,GA,PI)
2*      C
3*      C      ADJUSTABLE DIMENSIONS
4*      C
5*      INTEGER H(1),PI(1)
6*      REAL MB(1),MA(NH,7),PB(NH,3),PA(NH,NH,3),G(NH,3)
7*      PARAMETER Q=4
8*      PARAMETER S=Q+1,V=Q+3,VQ=4*V,S3=3*S
9*      C
10*     C      ADDITIONAL DIMENSIONED VARIABLES
11*     C
12*     DOUBLE PRECISION A(V,V1),WRK1(V4),BMASS(S)
13*     DIMENSION PH1S,3,31,DX1S,S1,DY1S,S1,DZ1S,S1,DX0(S,S1),DY0(S,S1),DZ0(S
14*     S1),VLX(S,S1),VLY(S,S1),LZ(S,S1),FEX0(S1),FEY0(S1),FEZ0(S1),GO(8,31),TX0(S
15*     S1),TY0(S1),TZ0(S1),PSIS,S,3,3),AS(8,Q),AV1Q,31,T(1,S,3,3),A0013,31,AB
16*     S(3,3),AIS131,EA(31),CH(31),CE(31)
17*     INTEGER KK(K)
18*     REAL LX,LY,LZ,IXX(S1),IYY(S1),IZZ(S1),IXY(S1),IXZ(S1),IYZ(S1)
19*     EQUIVALENCE (A,PS1),(LX,DX0,AS1),(LY,DY0),(LZ,DZ0)
20*     NE=NH+1
21*     C
22*     C      INITIALIZE EPSIK,J1 WITH ZEROS AND DEFINE KK(K)
23*     C
24*     DO 85 K=1,NH
25*     KK(K)=K+1
26*     DO 85 J=1,NB
27*     85 EPSIK,J1=0
28*     C
29*     C      DEFINE EPSIK,J1 USING H1J
30*     C
31*     DO 86 K=1,NH
32*     DO 86 J=2,NB
33*     IF(K.EQ.(J-1)) EPSIK,J1=1
34*     IF(K.LT.(J-1)) GO TO 87
35*     GO TO 86
36*     87 CONTINUE
37*     JK=K+1
38*     J1=-1
39*     DO 89 L=JK,J1
40*     IF(K.GT.(L-1)) GO TO 89
41*     IF((EPSIK,L).EQ.1).AND.(H1(J-1),EQ,(L-1))) EPSIK,J1=1
42*     89 CONTINUE
43*     86 CONTINUE
44*     C
45*     C      INITIAL CALCULATION OF BARYCENTER VECTORS W.R.T. BODY C.G.S
46*     C      AND HINGE POINTS
47*     C
48*     IXX(1)=MB(1)
49*     IYY(1)=MB(2)
50*     IZZ(1)=MB(3)
51*     IXY(1)=MB(4)
52*     IXZ(1)=MB(5)
53*     IYZ(1)=MB(6)
54*     BMASS(1)=MB(7)
55*     TM=BMASS(1)
56*     DO 35 J=2,NB
57*     IXX(J)=MA(J-1,1)
58*     IYY(J)=MA(J-1,2)
59*     IZZ(J)=MA(J-1,3)
60*     IXY(J)=MA(J-1,4)
61*     IXZ(J)=MA(J-1,5)
62*     IYZ(J)=MA(J-1,6)
63*     BMASS(J)=MA(J-1,7)
64*     35 TM=TM+BMASS(J)
65*     DO 149 I=1,NB

```

```

66*           I1=I-1
67*           DC 149 J=1,NB
68*           J1=J-1
69*           IF(I,EG,J) GO TO 163
70*           IF(I,GT,J) GO TO 70
71*           IF(I,EG,1) GO TO 80
72*           IF(EPS(I1,J),EQ,1) GO TO 400
73*           70   LX(I,J)=PA(I1,I1+1)
74*           LY(I,J)=PA(I1,I1+2)
75*           LZ(I,J)=PA(I1,I1+3)
76*           GO TO 149
77*           400  CONTINUE
78*           DO 600 K=I,J1
79*           IF(EPS(K,J),EQ,1) GO TO 500
80*           600  CONTINUE
81*           GC TO 149
82*           500  LX(I,J)=PA(I1,K+1)
83*           LY(I,J)=PA(I1,K+2)
84*           LZ(I,J)=PA(I1,K+3)
85*           GO TO 149
86*           80   DO 80 L=1,J1
87*           IF(EPS(L,J),EQ,1) GO TO 101
88*           90   CONTINUE
89*           GC TO 149
90*           101  LX(I,J)=PB(L+1)
91*           LY(I,J)=PB(L+2)
92*           LZ(I,J)=PB(L+3)
93*           GO TO 149
94*           163  LX(I,J)=0.
95*           LY(I,J)=0.
96*           LZ(I,J)=0.
97*           149  CONTINUE
98*           DO 13 N=1,NB
99*           DC 13 J=1,NB
100*          DX(N,J)=LX(N,J)
101*          DY(N,J)=LY(N,J)
102*          DZ(N,J)=LZ(N,J)
103*          DO 13 K=1,NB
104*          DX(N,J)=DX(N,J)-(BMASS(K)/TM)*LX(N,K)
105*          DY(N,J)=DY(N,J)-(BMASS(K)/TM)*LY(N,K)
106*          13   DZ(N,J)=DZ(N,J)-(BMASS(K)/TM)*LZ(N,K)
107*          C
108*          C   CALCULATION OF AUGMENTED INERTIA DYADICS FOR EACH BODY
109*          C
110*          DC 31 N=1,NB
111*          PH(N+1,1)=IXX(N)
112*          PH(N+1,2)=-IXY(N)
113*          PH(N+1,3)=-IXZ(N)
114*          PH(N+2,2)=IYY(N)
115*          PH(N+2,3)=-IYZ(N)
116*          PH(N+3,3)=IZZ(N)
117*          DO 30 J=1,NB
118*          PH(N+1,1)=PH(N+1,1)+BMASS(J)*(DY(N,J)**2+DZ(N,J)**2)
119*          PH(N+1,2)=PH(N+1,2)-BMASS(J)*DX(N,J)*DY(N,J)
120*          PH(N+1,3)=PH(N+1,3)-BMASS(J)*DX(N,J)*DZ(N,J)
121*          PH(N+2,2)=PH(N+2,2)+BMASS(J)*(DX(N,J)**2+DZ(N,J)**2)
122*          PH(N+2,3)=PH(N+2,3)-BMASS(J)*DX(N,J)*DZ(N,J)
123*          30   PH(N+3,3)=PH(N+3,3)+BMASS(J)*(DX(N,J)**2+DY(N,J)**2)
124*          PH(N+2,1)=PH(N+1,2)
125*          PH(N+3,1)=PH(N+1,3)
126*          31   PH(N+3,2)=PH(N+2,3)
127*          RETURN
128*          EMTRY MRATEINH,TH,IB,TB,FB,FA,GM,GMD,GMOD,WG,ECI
129*          DIMENSION TB(3),TA(NH,3),FB(3),FA(NH,3),GM(1),GMD(1),GMOD(1),
130*          1TH(NH),WD(3),WXC(S),WYO(S),WZO(S),E(S3,1)
131*          DOUBLE PRECISION ECI,V
132*          C
133*          C   BODY-TO-BODY COORDINATE TRANSFORMATION MATRICES
134*          C
135*          DO 335 J=1,NH
136*          K=H(J)

```

```

137*      SGM=SIN(GM(J))
138*      CGM=CCOS(GM(J))
139*      CGM1=1.-CGM
140*      G1=CGM1*G(J+1)
141*      G2=CGM1*G(J+2)
142*      G3=CGM1*G(J+3)
143*      SG1=SGM*G(J+1)
144*      SG2=SGM*G(J+2)
145*      SG3=SGM*G(J+3)
146*      G1S=G1*G(J+1)
147*      G2S=G2*G(J+2)
148*      G3S=G3*G(J+3)
149*      G12=G1*G(J+2)
150*      G13=G1*G(J+3)
151*      G23=G2*G(J+3)
152*      AB(1,1)=CGM+G15
153*      AB(1,2)=SG3+G12
154*      AB(1,3)=-SG2+G13
155*      AB(2,1)=-SG3+G12
156*      AB(2,2)=CGM+G25
157*      AB(2,3)=SG1+G23
158*      AB(3,1)=SG2+G13
159*      AB(3,2)=-SG1+G23
160*      AB(3,3)=CGM+G35
161*      IF(IK.EQ.0) GO TO 3350
162*      DO 334 L=1,3
163*      DC 334 M=1,3
164*      T(1,J+1,L,M)=0.
165*      DC 334 I=1,3
166*      334 T(1,J+1,L,M)=T(1,J+1,L,M)+AB(L,I)*T(I,K+1,I,M)
167*      GC TO 335
168*      3350 CONTINUE
169*      DC 3351 L=1,3
170*      DO 3351 M=1,3
171*      3351 T(1,J+1,L,M)=AB(L,M)
172*      335 CONTINUE
173*      C
174*      C      COORD. TRANSFORMATION OF G VECTORS (TO REF. BODY FRAME)
175*      C
176*      DO 362 I=1,NH
177*      DC 362 J=1,3
178*      GO(I,J)=0.
179*      DC 362 K=1,3
180*      GO(I,J)=GO(I,J)+T(1,I+1,K,J)*G(I,K)
181*      362 CONTINUE
182*      C
183*      C      ANG. VELOCITY COMPONENTS OF EACH BODY (IN REF. BODY FRAME)
184*      C
185*      DC 361 J=1,NB
186*      WX0(I,J)=W0(I,1)
187*      WY0(I,J)=W0(I,2)
188*      WZ0(I,J)=W0(I,3)
189*      DC 36 K=1,NH
190*      IF(LEPS(K,J).EQ.0) GO TO 36
191*      WX0(I,J)=WX0(I,J)+GMD(K)*GO(K+1)
192*      WY0(I,J)=WY0(I,J)+GMD(K)*GO(K+2)
193*      WZ0(I,J)=WZ0(I,J)+GMD(K)*GO(K+3)
194*      36 CONTINUE
195*      361 CONTINUE
196*      C
197*      C      TRANSFORM AUGMENTED BODY INERTIA DYNAMICS TO REF. BODY FRAME
198*      C
199*      DC 363 I=2,NB
200*      DO 364 J=1,3
201*      DC 364 K=1,3
202*      AB(J,K)=0.
203*      DC 364 L=1,3
204*      AB(J,K)=AB(J,K)+PHI(I,J,L)*T(I,I+L,K)
205*      364 CONTINUE
206*      DO 365 J=1,3
207*      DC 365 K=1,3
208*      PS(I,I,J,K)=0.

```

```

209*      DC 365 L=1,3
210*      PS(I,I,J,K)=PS(I,I,J,K)+T(I,I,L,J)*AB(L,K)
211*      365 CONTINUE
212*      363 CONTINUE
213*      FEXC(1)=FB(1)
214*      FEYC(1)=FB(2)
215*      FEZO(1)=FB(3)
216*      TXC(1)=TB(1)
217*      TYO(1)=TB(2)
218*      TZC(1)=TB(3)
219*      DO 17 I=2,NB
220*      K:I-1
221*      FEXO(I)=T(I,I,1,1)+FA(K,1)+T(I,I,2,1)*FA(K,2)+T(I,I,3,1)*FA(K,3)
222*      FEYC(I)=T(I,I,1,2)+FA(K,1)+T(I,I,2,2)*FA(K,2)+T(I,I,3,2)*FA(K,3)
223*      FEZO(I)=T(I,I,1,3)+FA(K,1)+T(I,I,2,3)*FA(K,2)+T(I,I,3,3)*FA(K,3)
224*      TXC(I)=T(I,I,1,1)+TA(K,1)+T(I,I,2,1)*TA(K,2)+T(I,I,3,1)*TA(K,3)
225*      TYO(I)=T(I,I,1,2)+TA(K,1)+T(I,I,2,2)*TA(K,2)+T(I,I,3,2)*TA(K,3)
226*      TZC(I)=T(I,I,1,3)+TA(K,1)+T(I,I,2,3)*TA(K,2)+T(I,I,3,3)*TA(K,3)
227*      DO 17 J=1,NB
228*      DXC(I,J)=T(I,I,1,1)*DX(I,J)+T(I,I,2,1)*DY(I,J)+T(I,I,3,1)*DZ(I,J)
229*      DYQ(I,J)=T(I,I,1,2)*DX(I,J)+T(I,I,2,2)*DY(I,J)+T(I,I,3,2)*DZ(I,J)
230*      DZC(I,J)=T(I,I,1,3)*DX(I,J)+T(I,I,2,3)*DY(I,J)+T(I,I,3,3)*DZ(I,J)
231*      17 CONTINUE
232*      DC 367 I=1,NB
233*      DXO(1,I)=DX(1,I)
234*      DYC(1,I)=DY(1,I)
235*      367 DZO(1,I)=DZ(1,I)
236*      C ADDITIONAL AUGMENTED INERTIA DYADICS (IN REF. BODY FRAME)
237*      C
238*      DO 37 I=1,NB
239*      DC 37 J=1,NB
240*      IF(I,EQ.J) GO TO 37
241*      PS(I,J,1,1)=-TM*(DYC(I,J)*DYQ(J,I)+DZO(I,J)*DZQ(J,I))
242*      PS(I,J,1,2)=IM*DXQ(J,I)*DYQ(I,J)
243*      PS(I,J,1,3)=TM*DXQ(I,J)*DZO(I,J)
244*      PS(I,J,2,1)=TM*DYO(I,J)*DXQ(I,J)
245*      PS(I,J,2,2)=-TM*(DXQ(I,J)*DXQ(J,I)*DZO(I,J)*DZQ(J,I))
246*      PS(I,J,2,3)=TM*DYO(I,J)*DZO(I,J)
247*      PS(I,J,3,1)=TM*DZQ(J,I)*DXQ(I,J)
248*      PS(I,J,3,2)=IM*DZO(J,I)*DYQ(I,J)
249*      PS(I,J,3,3)=-TM*(DXQ(I,J)*DXQ(J,I)*DYQ(I,J)*DZO(I,J))
250*      37 CONTINUE
251*      37
252*      DC 751 M=1,3
253*      DO 751 N=1,3
254*      751 PS(1,1,M,N)=PH(1,M,N)
255*      C VECTOR CROSS PRODUCTS DESCRIBING SYSTEM ROTATIONAL COUPLING.
256*      C (QUADRATIC TERMS INVOLVING THE CONNECTING BODY ANGULAR
257*      C VELOCITIES AND THE MUTUAL BARYCENTER-HINGE VECTORS)
258*      C
259*      DC 230 N=1,NB
260*      CPX=0.
261*      CFY=0.
262*      CPZ=0.
263*      DC 2301 L=1,NB
264*      IF(N.EQ.L) GO TO 2301
265*      WEX=WYO(L)*DZO(L,N)-WZQ(L)*DYQ(L,N)
266*      WDY=WZQ(L)*DXQ(L,N)-WXQ(L)*DZO(L,N)
267*      WEZ=WXQ(L)*DYO(L,N)-WYO(L)*DXQ(L,N)
268*      WWDX=TM*(WYO(L)*WDZ-WZQ(L)*WDY)*FEXO(L)
269*      WWDY=TM*(WZQ(L)*WDX-WXQ(L)*WDZ)*FEYC(L)
270*      WWDZ=TM*(WXQ(L)*WDY-WYO(L)*WDX)*FEZO(L)
271*      DWWDX=DYO(N,L)*WDZ-DZO(N,L)*WDY
272*      DWWDY=DZO(N,L)*WDX-DXQ(N,L)*WDZ
273*      DWWDZ=DXQ(N,L)*WDY-DYO(N,L)*WDX
274*      CFX=CPX+DWWDX
275*      CPY=CPY+DWWDY
276*      CFZ=CPZ+DWWDZ
277*      2301 CONTINUE
278*      DFX=DYC(N,N)*FEZO(N)-DZO(N,N)*FEYC(N)

```

```

280*      DF Y=DZO(N,N)+FEXO(N)-DXO(N,N)+FEZO(N)
281*      DFZ=DXO(N,N)+FEYO(N)-DVO(N,N)*FEXO(N)
282*      HX=PS(N,N+1)+WXO(N)*PS(N,N+1,2)+HYO(N)*PS(N,N+1,3)+WZO(N)
283*      HY=PS(N,N+2)+WXO(N)*PS(N,N+2,2)+HYO(N)*PS(N,N+2,3)+WZO(N)
284*      HZ=PS(N,N+3)+WXO(N)*PS(N,N+3,2)+HYO(N)*PS(N,N+3,3)+WZO(N)
285*      K = 3*(N-1)
286*      E1(K+1,1)=HY*WZO(N)-HZ*HYO(N)+TXO(N)+CPX+DFX
287*      E1(K+2,1)=HZ*WXO(N)-HX*WZO(N)+TYO(N)+CPY+DFY
288*      E1(K+3,1)=HX*HYO(N)-HY*WXO(N)+TZO(N)+CPZ+DFZ
289*      230 C(NTINUE
290*      C
291*      C      ACO MATRIX ELEMENT COMPUTATION {3X3}
292*      C
293*      DC 3001 I=1,3
294*      DO 3001 J=1,3
295*      3001 ACO(I,J)=0.
296*      DO 3 I=1,NB
297*      DC 3 J=1,NB
298*      AOO(I,1)=AOO(I,1)+PS(I,J,1,1)
299*      ACO(I,2)=AOO(I,2)+PS(I,J,1,2)
300*      AOO(I,3)=AOO(I,3)+PS(I,J,1,3)
301*      ACO(I,2,2)=AOO(I,2,2)+PS(I,J,2,2)
302*      AOO(I,2,3)=AOO(I,2,3)+PS(I,J,2,3)
303*      ACO(I,3,3)=AOO(I,3,3)+PS(I,J,3,3)
304*      3 CONTINUE
305*      ACO(2,1)=ACO(1,2)
306*      AOO(3,1)=AOO(1,3)
307*      ACO(3,2)=ACO(2,3)
308*      C
309*      C      ACK VECTOR ELEMENT COMPUTATION {3X1}
310*      C
311*      DC 7 K=1,NH
312*      AV(K,1)=0.
313*      AV(K,2)=0.
314*      AV(K,3)=0.
315*      DC 7 I=1,NB
316*      DO 7 J=1,NB
317*      IF(EPS(K,J).EQ.0) GO TO 7
318*      DO 10 M=1,3
319*      DC 10 L=1,3
320*      10 AVIK,M)=AVIK,M)+PS(I,J,M,L)+GOIK,L)
321*      7 C(NTINUE
322*      C
323*      C      APR SCALAR ELEMENT COMPUTATION
324*      C
325*      DC 14 K=1,NH
326*      DO 14 M=1,NH
327*      IF(K.GT.M) GO TO 14
328*      AIS(I)=0.
329*      AIS(2)=0.
330*      AIS(3)=0.
331*      DC 15 J=1,NB
332*      DO 15 I=1,NB
333*      IF(EPS(I,J).EQ.0).OR.(EPS(M,I).EQ.0) GO TO 15
334*      DO 18 N=1,3
335*      DC 18 L=1,3
336*      18 AIS(N)=AIS(N)+PS(J,I,N,L)+GOI,M,L)
337*      15 C(NTINUE
338*      14 AIS(K,M)=GOIK,1)+AIS(1)+GOIK,2)+AIS(2)+GOIK,3)+AIS(3)
339*      14 C(NTINUE
340*      C
341*      C      COMPUTE CORRECTION ELEMENTS FOR IEI VECTOR
342*      C
343*      DC 40 I=1,NB
344*      EA(1)=0.
345*      EA(2)=0.
346*      EA(3)=0.
347*      DC 41 J=1,NB
348*      DO 411 M=1,3
349*      411 EA(M)=0.
350*      DO 42 K=1,NH

```

```

351*      IF(EPS(K,J).EQ.0) GO TO 42
352*      KB=KK(K)
353*      CW(1)=CW(1)+GMD(K)*(WY0(KB)*GO(K,3)-WZC(KB)*GO(K,2))
354*      CW(2)=CW(2)+GMD(K)*(WZD(KB)*GO(K,1)-WXO(KB)*GO(K,3))
355*      CW(3)=CW(3)+GMD(K)*(WXO(KB)*GO(K,2)-WYC(KB)*GO(K,1))
356*      42 CONTINUE
357*      DC 45 M=1,3
358*      DO 45 L=1,3
359*      45 EA(M)=EA(M)+PS(I,J,M,L)*CW(L)
360*      41 CONTINUE
361*      K1=3*(I-1)
362*      EIK1+1,1=EIK1+1,1-EA(1)
363*      EIK1+2,1=EIK1+2,1-EA(2)
364*      EIK1+3,1=EIK1+3,1-EA(3)
365*      40 CONTINUE
366*      DO 55 M=1,3
367*      55 EC(M)=E(M,I)
368*      DO 52 J=2,NB
369*      DC 52 M=1,3
370*      K1=3*(J-1)+M
371*      52 EC(M)=EC(M)+EIK1,1
372*      I=0
373*      DC 60 K=1,NH
374*      IF(PI(IK).NE.0) GO TO 60
375*      I=I+1
376*      EC(I+3)=0.
377*      DC 601 M=1,3
378*      601 CE(M)=0.
379*      DC 61 J=1,NB
380*      IF(EPS(IK,J).EQ.0) GO TO 61
381*      DC 65 M=1,3
382*      J1=3*(J-1)+M
383*      65 CE(M)=CE(M)+E(J1,1)
384*      61 CONTINUE
385*      DO 66 L=1,3
386*      66 EC(I+3)=EC(I+3)+GO(IK,L)*CE(L)
387*      EC(I+3)=EC(I+3)+THIK
388*      60 CONTINUE
389*      DO 610 I=1,3
390*      DC 610 J=1,NH
391*      IF(PI(IJ).EQ.0) GO TO 610
392*      EC(I)=EC(I)-AV(J,I)*GMDD(I)
393*      610 CONTINUE
394*      K=0
395*      DO 612 I=1,NH
396*      IF(PI(I).NE.0) GO TO 612
397*      K=K+1
398*      I=K+3
399*      DO 611 J=1,NH
400*      IF(PI(I).EQ.0) GO TO 611
401*      IF(I.GT.J) AS(I,J)=AS(J,I)
402*      EC(K+3)=EC(K+3)-AS(I,J)*GMDD(I)
403*      611 CONTINUE
404*      612 CONTINUE
405*      C
406*      C   LOAD SYSTEM MATRIX (A) WITH ADD+ADK+AKH ELEMENTS
407*      C
408*      DC 23 I=1,3
409*      DO 23 J=1,3
410*      23 A(I,J)=ADD(I,J)
411*      DO 24 I=1,3
412*      K=0
413*      DO 24 J=1,NH
414*      IF(PI(J).NE.0) GO TO 24
415*      K=K+1
416*      A(K+3,I)=AV(J,I)
417*      A(I,K+3)=AV(J,I)
418*      24 CONTINUE
419*      K=0
420*      DC 250 I=1,NH
421*      IF(PI(I).NE.0) GO TO 250

```

```

422*      K=K+1
423*      L=0
424*      DC 25 J=1,NR
425*      IF IPI(J).NE.01 GO TO 25
426*      L=L+1
427*      IF(K.GT.L) GO TO 26
428*      A(IK+3,L+3)=AST(I,J)
429*      GO TO 25
430*      26 A(IK+3,L+3)=A(IL+3,K+3)
431*      25 CONTINUE
432*      25D CCNTINUE
433*      C
434*      C      SOLVE SYSTEM MATRIX FOR REFERENCE BODY ANG. ACCELERATION AND HINGE
435*      C          IRELATIVE) ROTATIONAL ACCELERATIONS
436*      C
437*      CALL SODIA,V,IV,EC,V,1,$92,WRK)
438*      DC 910 J=V+4
439*      K=J-3
440*      IF IPI(K).NE.01 GO TO 911
441*      EC(J)=EC(IV)
442*      IV=IV-1
443*      GC TO 910
444*      911 EC(J)=GMDO(K)
445*      910 CCNTINUE
446*      92 CONTINUE
447*      RETURN
448*      END

```

Appendix B

Summary of MBDYTL, MBDYL User Requirements

Subroutine Entry Statements

```
CALL MBDYTL(NH,H,MB,MA,PB,PA,G,PI)
or CALL MBDYL(NH,H,MB,MA,FB,PA,G,PI)
```

```
CALL MRATE(NH,TH,TB,TA,FB,FA,GM,GMD,GMDD,WO,WDOT)
```

Input/Output Variable Type and Storage Specifications—Fortran V

INTEGER NH, H(*n*), PI(*n*)

REAL MB(7), MA(*n*,7), PB(*n*,3), PA(*n,n*,3), G(*n*,3), TH(*n*),
TB(3), TA(*n*,3), FB(3), FA(*n*,3), GM(*n*), GMD(*n*),
GMD(*n*), WO(3)

DOUBLE PRECISION WDOT(*n* + 3)(where *n* = number of system hinges)

External Subroutines Called

AINVD — double precision matrix inversion subroutine; user-supplied (see line 224 in MBDYTL listing, line 223 in MBDYL listing)

This subroutine inverts the matrix *A*, where *A* is of order IV × IV,
 $IV \geq 3$. The result is returned in *A*.

Subroutine Setup

Insert the Fortran V statement (see listings)

PARAMETER Q = *n*

where *n* = number of system hinges

Data Restrictions

$n \geq 1$

Core Storage Required

Code: 1500 words

Data: 500 words (minimum; increases with *n*)

Listings

```

1*      SUBROUTINE MBODY(LINH,H,MB,MA,PB,PA,G,PI)
2*      C
3*      C      ADJUSTABLE DIMENSIONS
4*      C
5*      INTEGER H(1),PI(1)
6*      REAL MB(1),MA(NH,7),PB(NH,3),PA(NH,NH+3),G(NH,3)
7*      PARAMETER Q=4
8*      PARAMETER S=Q+1,V=Q+3,V4=S*V,S3=S*S
9*      C
10*     C      ADDITIONAL DIMENSIONED VARIABLES
11*     C
12*     DOUBLE PRECISION A(V,V),WRK(V4),BMASS(S)
13*     DIMENSION PH(S,3,3),DX(S,S),DY(S,S),DZ(S,S),DX0(S,S),DY0(S,S),
14*     DZ0(S,S),GU(Q,3),T1(S,3,3),LX(S,S),LY(S,S),LZ(S,S),FEX0(S),
15*     FSY0(S),FEZ0(S),IXG(S),IY0(S),TZ0(S),AB(3,3),
16*     ECE(3),AIS(3),PS(S,S,3,3),AS(0,Q),AVIQ(3),
17*     A0G(3,3)
18*     INTEGER KK(IQ),EPSIQ(S)
19*     REAL LX,LY,LZ,IXX(S),IYY(S),IZZ(S),IXY(S),IXZ(S),IYZ(S)
20*     EQUIVALENCE (PS,A),(LX,DX0),(LY,DY0),(LZ,DZ0),(PH,T)
21*     NB=NH+1
22*     C
23*     C      INITIALIZE EPSIK,J1 WITH ZEROS AND DEFINE KKIK
24*     C
25*     DO 85 K=1,NH
26*     KK(K)=K+1
27*     DO 85 J=1,NB
28*     EPSIK(J)=0
29*     C
30*     C      DEFINE EPSIK,J1 USING H(IJ)
31*     DO 86 K=1,NH
32*     DO 86 J=2,NB
33*     IF(.EQ.(J-1)) EPSIK(J)=1
34*     IF(K.LT.(J-1)) GO TO 87
35*     GO TO 86
36*     87  CONTINUE
37*     JK=K+1
38*     JI=J-1
39*     DO 88 L=J0,JI
40*     IF(K.GT.(L-1)) GO TO 88
41*     IF((EPSIK(K-1).EQ.1).AND.(H(J-1).EQ.(L-1))) EPSIK(K)=1
42*     88  CONTINUE
43*     86  CONTINUE
44*     C
45*     C      INITIAL CALCULATION OF BARYCENTER VECTORS W.R.T. BODY C.G.S
46*     C      AND HINGE POINTS
47*     C
48*     IXX(1)=MB(1)
49*     IYY(1)=MB(2)
50*     IZZ(1)=MB(3)
51*     IXY(1)=MB(4)
52*     IXZ(1)=MB(5)
53*     IYZ(1)=MB(6)
54*     BMASS(1)=MB(7)
55*     TM=BMASS(1)
56*     DO 35 J=2,NB
57*     IXX(J)=MA(J-1,1)
58*     IYY(J)=MA(J-1,2)
59*     IZZ(J)=MA(J-1,3)
60*     IXY(J)=MA(J-1,4)
61*     IXZ(J)=MA(J-1,5)
62*     IYZ(J)=MA(J-1,6)
63*     BMASS(J)=MA(J-1,7)
64*     35  TM=TM+BMASS(J)
65*     DO 149 I=1,NB

```

```

66*      I1=I-1
67*      DO 149 J=1,NB
68*      J1=J-1
69*      IF(I.EQ.J) GO TO 1E3
70*      IF(I.GT.J) GO TO 7D
71*      IF(I.EQ.1) GO TO 80
72*      IF(EPS(I1,J).EQ.1) GO TO 400
73*      70   LX(I,J)=PA(I1,I1+1)
74*              LY(I,J)=PA(I1,I1+2)
75*              LZ(I,J)=PA(I1,I1+3)
76*              GO TO 149
77*      400   CONTINUE
78*      DO 600 K=I,J1
79*      IF(EPS(K,J).EQ.1) GO TO 500
80*      600   CONTINUE
81*              GO TO 149
82*      500   LX(I,J)=PA(I1,K+1)
83*              LY(I,J)=PA(I1,K+2)
84*              LZ(I,J)=PA(I1,K+3)
85*              GO TO 149
86*      80   DO 90 L=1,J1
87*      IF(EPS(L,J).EQ.1) GO TO 101
88*      90   CONTINUE
89*              GO TO 149
90*      101   LX(I,J)=PB(L,1)
91*              LY(I,J)=PB(L,2)
92*              LZ(I,J)=PB(L,3)
93*              GO TO 149
94*      163   LX(I,J)=0.
95*              LY(I,J)=0.
96*              LZ(I,J)=0.
97*      149   CONTINUE
98*      DO 13 N=1,NB
99*      DO 13 J=1,NB
100*      DX(N,J)=LX1N,J)
101*      DY(N,J)=LY(N,J)
102*      DZ(N,J)=LZ(N,J)
103*      DO 13 K=1,NB
104*      DX(N,J)=DX(N,J)-(BMASS(K)/TM)*LX(N,K)
105*      DY(N,J)=DY(N,J)-(BMASS(K)/TM)*LY(N,K)
106*      13   DZ(N,J)=DZ(N,J)-(BMASS(K)/TM)*LZ(N,K)
107*      C
108*      C      CALCULATION OF AUGMENTED INERTIA DYADICS FOR EACH BODY
109*      C
110*      DO 31 N=1,NB
111*      PH(N,1,1)=IXX(N)
112*      PH(N,1,2)=-IXY(N)
113*      PH(N,1,3)=-IXZ(N)
114*      PH(N,2,2)=IYY(N)
115*      PH(N,2,3)=-IYZ(N)
116*      PH(N,3,3)=IZZ(N)
117*      DO 30 J=1,NB
118*      PH(N,1,1)=PH(N,1,1)+BMASS(J)*((DY(N,J)**2+DZ(N,J)**2)
119*      PH(N,1,2)=PH(N,1,2)-BMASS(J)*DX(N,J)*DY(N,J)
120*      PH(N,1,3)=PH(N,1,3)-BMASS(J)*DX(N,J)*DZ(N,J)
121*      PH(N,2,2)=PH(N,2,2)+BMASS(J)*((DX(N,J)**2+DZ(N,J)**2)
122*      PH(N,2,3)=PH(N,2,3)-BMASS(J)*DY(N,J)*DZ(N,J)
123*      30   PH(N,3,3)=PH(N,3,3)+BMASS(J)*((DX(N,J)**2+DY(N,J)**2)
124*      PH(N,2,1)=PH(N,1,2)
125*      PH(N,3,1)=PH(N,1,3)
126*      31   PH(N,3,2)=PH(N,2,3)
127*      C
128*      C      ADDITIONAL AUGMENTED INERTIA DYADICS (IN BODY I FRAME)
129*      C
130*      DO 751 J=1,NB
131*      DO 751 M=1,3
132*      DO 751 N=1,3
133*      751   PS(J,J,M,N)=PH(J,M,N)
134*      DO 37 I=1,NB
135*      DO 37 J=1,NB
136*      IF(I.EQ.J) GO TO 37

```

```

137*      PS(I,J,1,1)=-TM*(DY(I,J)*DY(J,I)+DZ(I,J)*DZ(J,I))
138*      PS(I,J+1,2)=TM*DX(J,I)*DY(I,J)
139*      PS(I,J,1,3)=TM*DX(J,I)*DZ(I,J)
140*      PS(I,J+2,1)=TM*DY(J,I)*DX(I,J)
141*      PS(I,J,2,2)=-TM*(DX(I,J)*DX(J,I)+DZ(I,J)*DZ(J,I))
142*      PS(I,J+2,3)=TM*DY(J,I)*DZ(I,J)
143*      PS(I,J,3,1)=TM*DZ(J,I)*DX(I,J)
144*      PS(I,J+3,2)=TM*DZ(J,I)*DY(I,J)
145*      PS(I,J,3,3)=-TM*(DX(I,J)*DX(J,I)+DY(I,J)*DZ(J,I))
146*      . 37  CONTINUE
147*      C
148*      C      ADD MATRIX ELEMENT COMPUTATION (3X3)
149*      C
150*      DO 3 I=1,NB
151*      DO 3 J=1,NB
152*      AOO(1,1)=AOO(1,1)+PS(I,J,1,1)
153*      AOO(1,2)=AOO(1,2)+PS(I,J,1,2)
154*      AOO(1,3)=AOO(1,3)+PS(I,J,1,3)
155*      AOO(2,2)=AOO(2,2)+PS(I,J,2,2)
156*      AOO(2,3)=AOO(2,3)+PS(I,J,2,3)
157*      AOO(3,3)=AOO(3,3)+PS(I,J,3,3)
158*      3  CONTINUE
159*      AOO(2,1)=AOO(1,2)
160*      AOO(3,1)=AOO(1,3)
161*      AOO(3,2)=AOO(2,3)
162*      C
163*      C      ADD VECTOR ELEMENT COMPUTATION (3X1)
164*      C
165*      DO 7 K=1,NH
166*      AV(K,1)=0.
167*      AV(K,2)=U.
168*      AV(K,3)=U.
169*      DO 7 I=1,NB
170*      DO 7 J=1,NB
171*      IF(EPS(K,J),EQ,0.) GO TO 7
172*      DO 10 M=1,3
173*      DO 10 L=1,3
174*      10  AV(K,M)=AV(K,M)+PS(I,J,M,L)*G(K,L)
175*      7  CONTINUE
176*      C
177*      C      AKM SCALAR ELEMENT COMPUTATION
178*      C
179*      DO 14 K=1,NH
180*      DO 14 M=1,NH
181*      IF(K.GT.M) GO TO 14
182*      AIS(1)=0.
183*      AIS(2)=0.
184*      AIS(3)=0.
185*      DO 15 J=1,NB
186*      DO 15 I=1,NB
187*      IF((EPS(K,J),EQ,0.),OR.,(EPS(M,I),EQ,0.)) GO TO 15
188*      DO 18 N=1,3
189*      DO 18 L=1,3
190*      18  AIS(N)=AIS(N)+PS(J,I,N,L)*G(M,L)
191*      15  CONTINUE
192*      AIS(K,M)=G(K,1)*AIS(1)+G(K,2)*AIS(2)+G(K,3)*AIS(3)
193*      14  CONTINUE
194*      C
195*      C      LOAD SYSTEM MATRIX (A) WITH AOO+ACK+AKM ELEMENTS
196*      C
197*      DO 23 I=1,3
198*      DO 23 J=1,3
199*      23  A(I,J)=AOO(I,J)
200*      DO 24 I=1,3
201*      K=0
202*      IV=3
203*      DO 24 J=1,NH
204*      IF(PI(J),NE,0.) GO TO 24
205*      K=K+1
206*      A(K+3,I)=AV(J,I)
207*      A(I,K+3)=AV(J,I)

```

```

208*      24    CONTINUE
209*      K=0
210*      DO 250 I=1,NH
211*      IF(PI(I).NE.0) GO TO 250
212*      K=K+1
213*      IV=IV+1
214*      L=0
215*      DO 25 J=1,NH
216*      IF(PI(J).NE.0) GO TO 25
217*      L=L+1
218*      IF(K.GT.L) GO TO 26
219*      A(L+3,L+3)=AS(I,J)
220*      GO TO 25
221*      26    A(K+3,L+3)=A(L+3,K+3)
222*      25    CONTINUE
223*      250   CONTINUE
224*      CALL AINVDT(A,V,IV,$1099,WRK)
225*      1099   CONTINUE
226*      RETURN
227*      ENTRY MRATE(NH,TH,TB,TA,FB,FA,GM,GMD,GMDD,WE,EQ)
228*      DIMENSION TB(3),TA(NH,3),FB(3),FA(NH,3),GM(1),GMD(1),GMDD(1),
229*      *TH(NH),WC(3)*E(3,1)
230*      DOUBLE PRECISION EC(V),EQ(V)
231*      C
232*      C      BODY-TO-BODY COORDINATE TRANSFORMATIONS
233*      C
234*      DO 335 J=1,NH
235*      K=H(J)
236*      AB(1,1)=1.
237*      AB(1,2)=GM(J)*G(J+3)
238*      AB(1,3)=-GM(J)*G(J+2)
239*      AB(2,1)=-AB(1,2)
240*      AB(2,2)=1.
241*      AB(2,3)=GM(J)*G(J+1)
242*      AB(3,1)=-AB(1,3)
243*      AB(3,2)=-AB(2,3)
244*      AB(3,3)=1.
245*      IF(K.EQ.0) GO TO 335C
246*      DO 334 L=1,3
247*      DO 334 M=1,3
248*      T(1,J+1,L,M)=0.
249*      DO 334 I=1,3
250*      334   T(1,J+1,L,M)=T(1,J+1,L,M)+AB(L,I)*T(1,K+1,I,M)
251*      GO TO 335
252*      3350  CONTINUE
253*      DO 3351 L=1,3
254*      DO 3351 M=1,3
255*      3351   T(1,J+1,L,M)=AB(L,M)
256*      335   CONTINUE
257*      C
258*      C      COORD. TRANSFORMATION OF G VECTORS (TO REF. BODY FRAME)
259*      C
260*      DO 362 I=1,NH
261*      DO 362 J=1,3
262*      GO(I,J)=0.
263*      DO 362 K=1,3
264*      GO(I,J)=GO(I,J)+T(1,I+1,K,J)*G(I,K)
265*      362   CONTINUE
266*      C
267*      C      EXTERNAL FORCES, EXTERNAL TORQUES
268*      C
269*      FEX0(I)=FB(1)
270*      FEY0(I)=FB(2)
271*      FEZ0(I)=FB(3)
272*      TX0(I)=TB(1)
273*      TY0(I)=TB(2)
274*      TZ0(I)=TB(3)
275*      DO 17 I=2,NB
276*      K=I-1
277*      FEX0(I)=T(1,I,1,1)*FA(K,1)+T(1,I+2,1)*FA(K,2)+T(1,I+3,1)*FA(K,3)
278*      FEY0(I)=T(1,I,1,2)*FA(K,1)+T(1,I+2,2)*FA(K,2)+T(1,I+3,2)*FA(K,3)

```

```

279*      FEZO(I)=T(1,I+1,3)*FA(K,1)+T(1,I+2,3)*FA(K,2)+T(1,I,3+3)*FA(K,3)
280*      TXO(I)=T(1,I,1,1)*TA(K,1)+T(1,I,2,1)*TA(K,2)+T(1,I,3,1)*TA(K,3)
281*      TYO(I)=T(1,I,1,2)*TA(K,1)+T(1,I,2,2)*TA(K,2)+T(1,I,3,2)*TA(K,3)
282*      TZO(I)=T(1,I,1,3)*TA(K,1)+T(1,I,2,3)*TA(K,2)+T(1,I,3,3)*TA(K,3)
283*      DO 17 J=1,NB
284*      DXO(I,J)=T(1,I,1,1)*DX(I,J)+T(1,I,2,1)*DY(I,J)+T(1,I,3,1)*DZ(I,J)
285*      DYO(I,J)=T(1,I,1,2)*DX(I,J)+T(1,I,2,2)*DY(I,J)+T(1,I,3,2)*DZ(I,J)
286*      DZO(I,J)=T(1,I,1,3)*DX(I,J)+T(1,I,2,3)*DY(I,J)+T(1,I,3,3)*DZ(I,J)
287*      17 CONTINUE
288*      DO 367 I=1,NB
289*      DXC(I,I)=DX(I,I)
290*      DYO(I,I)=DY(I,I)
291*      367 DZO(I,I)=DZ(I,I)
292*      C
293*      C VECTOR CROSS PRODUCTS DESCRIBING SYSTEM ROTATIONAL COUPLING.
294*      C {QUADRATIC TERMS INVOLVING THE CONNECTING BODY ANGULAR
295*      C VELOCITIES AND THE MUTUAL BARYCENTER-HINGE VECTORS}
296*      C
297*      DO 230 N=1,NB
298*      CPX=0.
299*      CPY=0.
300*      CPZ=0.
301*      DO 2301 L=1,NB
302*      CPX=CPX+DY(N,L)*FEZO(L)-DZO(N,L)*FEYO(L)
303*      CPY=CPY+DZO(N,L)*FEXO(L)-DXO(N,L)*FEZO(L)
304*      CPZ=CPZ+DXO(N,L)*FEYO(L)-DYO(N,L)*FEXO(L)
305*      2301 CONTINUE
306*      K = 3*(N-1)
307*      E(K+1,1)=TXO(K)+CPX
308*      E(K+2,1)=TYO(K)+CPY
309*      E(K+3,1)=TZO(K)+CPZ
310*      230 CONTINUE
311*      C
312*      C COMPUTE CORRECTION ELEMENTS FOR (E) VECTOR
313*      C
314*      DO 55 MI=1,3
315*      55 EC(MI)=E(MI+1)
316*      DO 52 J=2,NB
317*      DO 52 M=1,3
318*      K1=3*(J-1)+M
319*      52 EC(M)=EC(M)+E(K1+1)
320*      I=0
321*      DO 60 K=1,NH
322*      IF(PI(K).NE.0) GO TO 60
323*      I=I+1
324*      EC(I+3)=0.
325*      DO 601 M=1,3
326*      601 CE(M)=0.
327*      DO 61 J=1,NB
328*      IF(EPS(K+J).EQ.0) GO TO 61
329*      DO 65 M=1,3
330*      J1=3*(J-1)+M
331*      65 CE(M)=CE(M)+E(J1+1)
332*      61 CONTINUE
333*      DO 66 L=1,3
334*      66 EC(I+3)=EC(I+3)+GO(K,L)*CE(L)
335*      EC(I+3)=EC(I+3)+TH(K)
336*      60 CONTINUE
337*      DO 610 I=1,3
338*      DO 610 J=1,NH
339*      IF(PI(J).EQ.0) GO TO 610
340*      EC(I)=EC(I)-AV(J,I)*GMDD(J)
341*      610 CONTINUE
342*      K=0
343*      IV=3
344*      DO 612 I=1,NH
345*      IF(PI(I).NE.0) GO TO 612
346*      K=K+1
347*      IV=IV+1
348*      DO 611 J=1,NH
349*      IF(PI(J).EQ.0) GO TO 611

```

```

350*      IF(I.GT.J) AS(I,J)=AS(J,I)
351*      EC(K+3)=EC(K+3)-AS(I,J)*GMDD(J)
352*      611  CONTINUE
353*      612  CONTINUE
354*      C
355*      C      SOLVE SYSTEM MATRIX FOR REFERENCE BODY ANG. ACCELERATION AND WINGE
356*      C          (RELATIVE) ROTATIONAL ACCELERATIONS
357*      C
358*      DO 671 I=1,IV
359*      EQ(I)=0.
360*      DO 671 J=1,IV
361*      EQ(I)=EQ(I)+A(I,J)*EC(J)
362*      671  CONTINUE
363*      DO 910 J=V,4
364*      K=J-3
365*      IF(PI(K).NE.0) GO TO 911
366*      EQ(J)=EQ(IV)
367*      IV=IV-1
368*      GO TO 910
369*      911  EQ(J)=GMDD(K)
370*      910  CONTINUE
371*      92   CONTINUE
372*      RETURN
373*      END

```

```

1*      SUBROUTINE MBOYL(NH,H,MB,MA,PB,PA,G,PI)
2*      C
3*      C      ADJUSTABLE DIMENSIONS
4*      C
5*      INTEGER H1,I,PI(1)
6*      REAL MB(1),MA(NH,3),PB(NH,3),PA(NH,NH,3),G(NH,3)
7*      PARAMETER Q=4
8*      PARAMETER S=Q+1,V=Q+3,U=4*V,S3=3*S
9*      C
10*     C      ADDITIONAL DIMENSIONED VARIABLES
11*     C
12*     DOUBLE PRECISION A(V,V),WRK1(V4),BMASS(5)
13*     DIMENSION PHIS(3,31),DX(S,S),DY(S,S),DZ(S,S),
14*     GLX(S,S),LYIS,S1,LZ(S,S),FEX(S),FEY(S),FEZ(S),TX(S),TY(S),TZ(S),
15*     SCE(3),AIS(3),PS(S,S+3,31),AS10(0),AV(0,3),
16*     A00(3,3)
17*     INTEGER KK(0),EPS(0,S)
18*     REAL LX,LY,LZ,IYX(S),IYY(S),IZZ(S),IXY(S),IXZ(S),IYZ(S)
19*     EQUIVALENCE IPS,LX,A1
20*     N6=NH+1
21*     C
22*     C      INITIALIZE EPS(K,J) WITH ZEROS AND DEFINE KK(K)
23*     DO 85 K=1,NH
24*     KK(K)=K+1
25*     DO 85 J=1,NB
26*     EPS(K,J)=0
27*     C
28*     C      DEFINE EPS(K,J) USING H(J)
29*     C
30*     DO 85 K=1,NH
31*     DO 86 J=2,NB
32*     IF(K,E3.1J-1)) EPS(K,J)=1
33*     IF(K,LT,1J-1)) GO TO 87
34*     GO TO 86
35*     87    CONTINUE
36*     JO=K+1
37*     J1=J-1
38*     DO 89 L=JO,J1
39*     IF(IK,GT,1L-1)) GO TO 89
40*     IF((EPS(K-1,1).EQ.1),AND,(H(J-1).EQ.(L-1))) EPS(K,J)=1
41*     89    CONTINUE
42*     88    CONTINUE
43*     C
44*     C      INITIAL CALCULATION OF BARYCENTER VECTORS W.R.T. BODY C.G.S
45*     C      AND HINGE POINTS
46*     C
47*     IXX(1)=MB(1)
48*     IYY(1)=MB(2)
49*     IZZ(1)=MB(3)
50*     IXY(1)=MB(4)
51*     IXZ(1)=MB(5)
52*     IYZ(1)=MB(6)
53*     BMASS(1)=MB(7)
54*     TM=BMASS(1)
55*     DO 35 J=2,NB
56*     IXX(J)=MA(J-1,1)
57*     IYY(J)=MA(J-1,2)
58*     IZZ(J)=MA(J-1,3)
59*     IXY(J)=MA(J-1,4)
60*     IXZ(J)=MA(J-1,5)
61*     IYZ(J)=MA(J-1,6)
62*     BMASS(J)=MA(J-1,7)
63*     35    TM=TM+BMASS(J)
64*     DO 149 I=1,NB

```

```

65*      I1=I-1
66*      DO 149 J=1,NB
67*      J1=J-1
68*      IF(I.EQ.J) GO TO 153
69*      IF(I.GT.J) GO TO 70
70*      IF(I.EQ.1) GO TO 80
71*      IF(EPS(I).GT.J).EQ.I) GO TO 400
72*      70  LX(I,J)=PA(I,I,I+1)
73*      LY(I,J)=PA(I,I,I+1,2)
74*      LZ(I,J)=PA(I,I,I+1,3)
75*      GO TO 149
76*      400  CONTINUE
77*      DO 600 K=I,J1
78*      IF(EPS(K,J).EQ.1) GO TO 500
79*      600  CONTINUE
80*      GO TO 149
81*      500  LX(I,J)=PA(I,I,K+1)
82*      LY(I,J)=PA(I,I,K+2)
83*      LZ(I,J)=PA(I,I,K+3)
84*      GO TO 149
85*      80  DO 90 L=I,J1
86*      IF(EPS(L,J).EQ.1) GO TO 101
87*      90  CONTINUE
88*      GO TO 149
89*      101  LX(I,J)=PB(L,1)
90*      LY(I,J)=PB(L,2)
91*      LZ(I,J)=PB(L,3)
92*      GO TO 149
93*      103  LX(I,J)=0.
94*      LY(I,J)=0.
95*      LZ(I,J)=0.
96*      149  CONTINUE
97*      DO 13 N=1,NB
98*      DO 13 J=1,NB
99*      DX(N,J)=LX(N,J)
100*      DY(N,J)=LY(N,J)
101*      DZ(N,J)=LZ(N,J)
102*      DO 13 K=1,NB
103*      DX(N,J)=DX(N,J)-BMASS(K)/TM)*LX(N+K)
104*      DY(N,J)=DY(N,J)-BMASS(K)/TM)*LY(N+K)
105*      13  DZ(N,J)=DZ(N,J)-BMASS(K)/TM)*LZ(N+K)
106*      C
107*      C      CALCULATION OF AUGMENTED INERTIA DYADICS FOR EACH BODY
108*      C
109*      DO 31 N=1,NB
110*      PH(N,1,1)=IXX(N)
111*      PH(N,1,2)=-IXY(N)
112*      PH(N,1,3)=-IXZ(N)
113*      PH(N,2,2)=IYY(N)
114*      PH(N,2,3)=-IYZ(N)
115*      PH(N,3,3)=IZZ(N)
116*      DO 30 J=1,NB
117*      PH(N,1,1)=PH(N,1,1)+BMASS(J)*DX(N,J)**2+DZ(N,J)**2
118*      PH(N,1,2)=PH(N,1,2)-BMASS(J)*DX(N,J)*DY(N,J)
119*      PH(N,1,3)=PH(N,1,3)-BMASS(J)*DX(N,J)*DZ(N,J)
120*      PH(N,2,2)=PH(N,2,2)+BMASS(J)*DX(N,J)**2+DZ(N,J)**2
121*      PH(N,2,3)=PH(N,2,3)-BMASS(J)*DY(N,J)*DZ(N,J)
122*      30  PH(N,3,3)=PH(N,3,3)+BMASS(J)*DX(N,J)**2+DY(N,J)**2
123*      PH(N,2,1)=PH(N,1,2)
124*      PH(N,3,1)=PH(N,1,3)
125*      31  PH(N,3,2)=PH(N,2,3)
126*      C
127*      C      ADDITIONAL AUGMENTED INERTIA DYADICS (IN BODY I FRAME)
128*      C
129*      DO 751 J=1,NB
130*      DO 751 M=1,3
131*      DO 751 N=1,3
132*      751 PS(J,J,M,N)=PH(J,M,N)
133*      DO 37 I=1,NB
134*      DO 37 J=1,NB
135*      IF(I.EQ.J) GO TO 37

```

```

136*      PS(I,J+1,I)=-TM*(DY(I,J)*DY(J,I)+DZ(I,J)*DZ(J,I))
137*      PS(I,J+1,2)=TM*DX(J,I)*DY(I,J)
138*      PS(I,J+1,3)=TM*DX(J,I)*DZ(I,J)
139*      PS(I,J+2,I)=TM*DY(J,I)*DX(I,J)
140*      PS(I,J+2,2)=TM*DX(I,J)*DX(J,I)+DZ(I,J)*DZ(J,I)
141*      PS(I,J+2,3)=TM*DY(J,I)*DZ(I,J)
142*      PS(I,J+3,I)=TM*DZ(J,I)*DX(I,J)
143*      PS(I,J+3,2)=TM*DZ(J,I)*DY(I,J)
144*      PS(I,J+3,3)=TM*(DX(I,J)*DX(J,I)+DY(I,J)*DY(J,I))
145*      37  CONTINUE
146*      C
147*      C      ADD MATRIX ELEMENT COMPUTATION (3X3)
148*      C
149*      DO 3 I=1,NB
150*      DO 3 J=1,NB
151*      ADD(1,1)=ADD(1,1)+PS(I,J+1,I)
152*      ADD(1,2)=ADD(1,2)+PS(I,J+1,2)
153*      ADD(1,3)=ADD(1,3)+PS(I,J+1,3)
154*      ADD(2,1)=ADD(2,1)+PS(I,J+2,I)
155*      ADD(2,2)=ADD(2,2)+PS(I,J+2,2)
156*      ADD(2,3)=ADD(2,3)+PS(I,J+2,3)
157*      3  CONTINUE
158*      ADD(3,1)=ADD(3,1)+PS(I,J+3,I)
159*      ADD(3,2)=ADD(3,2)+PS(I,J+3,2)
160*      161*      C
162*      C      AGK VECTOR ELEMENT COMPUTATION (3X1)
163*      C
164*      DO 7 K=1,NH
165*      AV(K,1)=0.
166*      AV(K,2)=0.
167*      AV(K,3)=0.
168*      DO 7 I=1,NB
169*      DO 7 J=1,NB
170*      IF(EPS(IK,J),EQ,0) GO TO 7
171*      DO 10 M=1,3
172*      DO 10 L=1,3
173*      10  AV(K,M)=AV(K,M)+PS(I,J,M,L)*G(K,L)
174*      7  CONTINUE
175*      C
176*      C      AKM SCALAR ELEMENT COMPUTATION
177*      C
178*      DO 14 K=1,NH
179*      DO 14 M=1,NH
180*      IF(K,GT,M) GO TO 14
181*      AIS(1)=0.
182*      AIS(2)=0.
183*      AIS(3)=0.
184*      DO 15 J=1,NB
185*      DO 15 I=1,NB
186*      IF((EPS(IK,J),EQ,0).OR.(EPS(M,I),EQ,0)) GO TO 15
187*      DO 18 N=1,3
188*      DO 18 L=1,3
189*      18  AIS(N)=AIS(N)+PS(I,J,N,L)*G(M,L)
190*      15  CONTINUE
191*      15  AIS(K,M)=G(K,1)*AIS(1)+G(K,2)*AIS(2)+G(K,3)*AIS(3)
192*      14  CONTINUE
193*      C
194*      C      LOAD SYSTEM MATRIX (A) WITH ADD,AGK,AKM ELEMENTS
195*      C
196*      DO 23 I=1,3
197*      DO 23 J=1,3
198*      23  A(I,J)=ADD(I,J)
199*      DO 24 I=1,3
200*      K=0
201*      IV=3
202*      DO 24 J=1,NH
203*      IF(PI(J),NE,0) GO TO 24
204*      K=K+1
205*      A(K+3,I)=AV(I,J)
206*      A(I,K+3)=AV(J,I)

```

```

207*      24    CONTINUE
208*      K=0
209*      DO 250 I=1,NH
210*      IF(FI(I).NE.0) GO TO 250
211*      K=K+1
212*      IV=IV+1
213*      L=L0
214*      DO 25 J=1,NH
215*      IF(PI(J).NE.0) GO TO 25
216*      L=L+1
217*      IF(K.GT.L) GO TO 26
218*      A(K+3,L+3)=AS(I,J)
219*      GO TO 25
220*      26    A(K+3,L+3)=AL+3,K+3
221*      25    CONTINUE
222*      250   CONTINUE
223*      CALL AINV(A,V,IV,$1099,WRK)
224*      1099  CONTINUE
225*      RETURN
226*      ENTRY MRATE(INH,TH,TB,TA,FB,FA,GM,GMD,GMDD,WG,EG)
227*      DIMENSION TB(3),TA(NH,3),FB(3),FA(NH,3),GM(1),GMD(1),GMDD(1),
228*      $TH(NH),WG(3),EIS(3,1)
229*      DOUBLE PRECISION ECIV,EG(V)
230*
231*      C      EXTERNAL FORCES, EXTERNAL TORQUES, HINGE TORQUES
232*      C
233*      FE(X(1))=FB(1)
234*      FE(Y(1))=FB(2)
235*      FE(Z(1))=FB(3)
236*      TX(1)=TB(1)
237*      TY(1)=TB(2)
238*      TZ(1)=TB(3)
239*      DO 33 J=2,NB
240*      TX(J)=TA(J-1,1)
241*      TY(J)=TA(J-1,2)
242*      TZ(J)=TA(J-1,3)
243*      FE(X(J))=FA(J-1,1)
244*      FE(Y(J))=FA(J-1,2)
245*      33    FE(Z(J))=FA(J-1,3)
246*      C
247*      C      VECTOR CROSS PRODUCTS DESCRIBING SYSTEM ROTATIONAL COUPLING.
248*      C      QUADRATIC TERMS INVOLVING THE CONNECTING BODY ANGULAR
249*      C      VELOCITIES AND THE MUTUAL BARYCENTER-HINGE VECTORS
250*      C
251*      DO 230 N=1,NB
252*      CPX=0.
253*      CPY=0.
254*      CPZ=0.
255*      DO 2301 L=1,NB
256*      CPX=CPX+DY(N,L)*FEZ(L)-DZ(N,L)*FEY(L)
257*      CPY=CPY+DZ(N,L)*FEX(L)-DX(N,L)*FEZ(L)
258*      CPZ=CPZ+DX(N,L)*FEY(L)-DY(N,L)*FEX(L)
259*      2301  CONTINUE
260*      K = 3*IN-1
261*      E(K+1,1)=TX(N)+CPX
262*      E(K+2,1)=TY(N)+CPY
263*      E(K+3,1)=TZ(N)+CPZ
264*      230   CONTINUE
265*      C
266*      C      COMPUTE CORRECTION ELEMENTS FOR (E) VECTOR
267*      C
268*      DO 55 MI=1,3
269*      55    EC(MI)=E(MI,1)
270*      DO 52 J=2,NB
271*      DO 52 M=1,3
272*      K1=3*(J-1)+M
273*      52    EC(M)=EC(M)+E(K1,1)
274*      I=G
275*      DO 50 K=1,NH
276*      IF(PI(K).NE.0) GO TO 50
277*      I=I+1

```

```

278*      ECII+3)=0.
279*      DO E01 M=1,3
280*      CEIM)=0.
281*      DO E1 J=1,NB
282*      IF(EPSIK,J),EQ,0) GO TO 61
283*      DO E5 M=1,3
284*      J1=3*(J-1)+M
285*      65      CE(M)=CE(M)+E(J1,1)
286*      61      CONTINUE
287*      DO E6 L=1,3
288*      EC(I+3)=EC(I+3)+G(K,L)*CE(L)
289*      EC(I+3)=EC(I+3)+THIK
290*      60      CONTINUE
291*      DO E10 I=1,3
292*      DO S10 J=1,NH
293*      IF(P1(J),EQ,0) GO TO 610
294*      ECII)=EC(I)-AV(J,I)*GMDD(J)
295*      610     CONTINUE
296*      K=0
297*      IV=3
298*      DO S12 I=1,NH
299*      IF(P1(I),NE,0) GO TO E12
300*      K=K+1
301*      IV=IV+1
302*      DO S11 J=1,NH
303*      IF(P1(J),EQ,0) GO TO E11
304*      IF(I,GT,J) AS(I,J)=AS(J,I)
305*      EC(K+3)=EC(K+3)-ASII,J)*GMDD(J)
306*      611     CONTINUE
307*      612     CONTINUE
308*      C
309*      C      SOLVE SYSTEM MATRIX FOR REFERENCE BODY ANG. ACCELERATION AND HINGE
310*      C      (RELATIVE) ROTATIONAL ACCELERATIONS
311*      C
312*      DO 571 I=1,IV
313*      EQ(I)=0.
314*      DO 571 J=1,IV
315*      EG(I)=EG(I)+AF(I,J)*EC(J)
316*      671     CONTINUE
317*      DO S10 J=V+4
318*      K=J-3
319*      IF(P1(K),NE,0) GO TO 911
320*      EQ(J)=EQ(IV)
321*      IV=IV-1
322*      GO TO 910
323*      911     EQ(J)=GMDD(K)
324*      910     CONTINUE
325*      92      CONTINUE
326*      RETURN
327*      END

```

Appendix C

Summary of MBDYPL User Requirements

Subroutine Entry Statements

```
CALL MBDYPL(NH,H,MB,MA,PB,PA,G,PI,LI,TI)  
CALL MRATE(NH,TH,TB,TA,FB,FA,GM,GMD,GMDD,WO,WDOT)
```

Input/Output Variable Type and Storage Specifications—Fortran V

```
INTEGER NH, H(n), PI(n), LI(n), TI(n)  
REAL MB(7), MA(n,7), PB(n,3), PA(n,n,3), G(n,3), TH(n),  
      TB(3), TA(n,3), FB(3), FA(n,3), GM(n), GMD(n),  
      GMDD(n), WO(3)  
DOUBLE PRECISION WDOT(n + 3)(where n = number of sys-  
tem hinges)
```

External Subroutines Called

1. AINV—double precision matrix inversion subroutine; user-supplied (see line 272 in MBDYPL listing)

This subroutine inverts the matrix AT , where AT is of order $NMT \times NMT$, $NMT \geq 1$. The result is returned in AT .

2. SOD—double precision matrix equation solving subroutine; user-supplied (see line 666 in MBDYPL listing)

This subroutine solves the matrix equation $A\ddot{x} = EE$, where A is of order $NL \times NL$, $NL \geq 3$.

Subroutine Setup

Insert the Fortran V statement (see listing)

```
PARAMETER Q = n
```

where n = number of system hinges.

Data Restrictions

$n \geq 1$

Core Storage Required

Code: 3000 words

Data: 1000 words (minimum; increases with n)

Listing

```

1*      SUBROUTINE MBDYPL(NH,H,MB,MA,PB,PA,G,PI,L1,I1)
2*      C
3*      C      ADJUSTABLE DIMENSIONS
4*      C
5*      INTEGER H(1),PI(1),L1(1),I1(1)
6*      REAL MB(1),MA(NH,7),PB(NH,3),PA(NH,NH,3),G(NH,3)
7*      PARAMETER G=4
8*      PARAMETER U=G+1,V=G+3,W4=4*V+S3=3*U
9*      C
10*     C      ADDITIONAL DIMENSIONED VARIABLES
11*     C
12*     DOUBLE PRECISION AIV,UJ,WRK1(V4),BMASS(U)
13*     DOUBLE PRECISION AT1Q,Q1,AE(U,V),EE(V),ALIV,V
14*     DIMENSION PSU,U,3,31,DX(U,U),DY(U,U),DZ(U,U),
15*     $      LX(U,U),LY(U,U),LZ(U,U),PH1U,3,31
16*     DIMENSION AB(3,31),CE(31),CW(31),AIS(31),EA(31),G01Q,31,AD01(3,31),AV1Q,3
17*     $,AS1Q,Q1,T1U,3,31,BA(3,31),AS21Q,Q1,DX01U,U),DY01U,U),DZ01U,U),
18*     $TX01U),TY01U),TZ01U),FEX01U),FEY01U),FEZ01U)
19*     DIMENSION G1S(Q),G2S(Q),G3S(Q),G121Q),G131Q),G231Q)
20*     INTEGER KK1Q),EPS1Q,U1,R,S,TAM1Q),TAMF1Q),NUMF(U),TAB
21*     REAL LX,LY,LZ,IXX1U),IYY1U),IZZ1U),IXY1U),IXZ1U),IYZ1U)
22*     EQUIVALENCE (A,PS),(LX,DX0,AS),(LY,DY0),LZ,DZ0)
23*     NB=NH+1
24*     INT=V
25*     DO 506 K=1,NH
26*   506 LISUM=LISUM+L1(K)
27*     C
28*     C      INITIALIZE EPS(K,J) WITH ZEROS AND DEFINE KK1K
29*     C
30*     DO 85 K=1,NH
31*     KK1(K)=K+1
32*     DO 85 J=1,NB
33*   85 EPS1K,J)=0
34*     C
35*     C      DEFINE EPS1K,JI USING H(J)
36*     C
37*     DO 96 K=1,NH
38*     DO 96 J=2,NB
39*     IF1K,EQ,(J-1)) EPS1K,J)=1
40*     IF1K,LT,(J-1)) GO TO 87
41*     GO TO 86
42*   87  CONTINUE
43*     J0=K+1
44*     J1=J-1
45*     DO 89 L=JG,J1
46*     IF1K,GT,IL-1)) GO TO 89
47*     IF((EPS1K,L),EQ,1),AND,(H(J-1),EQ,(L-1)) EPS1K,J)=1
48*   89  CONTINUE
49*   86  CONTINUE
50*     C
51*     C      LOCATE POTENTIAL TERMINAL APPENDAGE BODIES
52*     C
53*     NN=0
54*     M=0
55*     LL=0
56*   805  CONTINUE
57*     DO 809 J=1,NH
58*     IF(H(J),EQ,M) GO TO 808
59*     GO TO 809
60*   808  LL=LL+1
61*     TAM(LL)=J
62*   809  CONTINUE
63*     DO 810 L=1,NH
64*     K=TAM(L)

```

```

65*      IF(TAMIL).EQ.0) GO TO 810
66*      NPI=0
67*      NEP=0
68*      DO 807 J=2,NB
69*      IF(EPSIK,J).EQ.1) NEP=NEP+1
70*      IF(LI(J-1).EQ.1).AND.(EPS(K,J).EQ.1)) NPI=NPI+1
71*      807 CONTINUE
72*      IF(NPI.EQ.0) GO TO 810
73*      IF(NPI.EQ.NEP) GO TO 820
74*      M=TAMIL
75*      TAMIL=0
76*      GO TO 805
77*      820 CONTINUE
78*      TAM(L)=0
79*      DO 827 J=2,NB
80*      IF(EPSIK,J).EQ.1) GO TO 828
81*      GO TO 827
82*      828 NN=NN+1
83*      TAMF(NN)=J-1
84*      827 CONTINUE
85*      810 CONTINUE
86*
87*      C LOCATE TERM. APPENDAGE BASE BODIES AND NO. OF T.A. BODIES ON EACH
88*      C
89*      DO 836 L=1,NB
90*      836 NUMFIL=0
91*      DO 835 L=1,NH
92*      DO 834 K=1,NH
93*      IF(TAMF(L).NE.K) GO TO 834
94*      N=HK+1
95*      DO 833 J=2,NB
96*      IF(EPSIK,J).EQ.0) GO TO 833
97*      NUMF(N)=NUMF(N)+1
98*      833 CONTINUE
99*      834 CONTINUE
100*     836 CONTINUE
101*     C
102*     C FIND BASE BODY WITH THE MOST TERMINAL APPENDAGE BODIES ON IT
103*     C
104*     NMX=0
105*     DO 837 L=1,NB
106*     IF(NUMLYL.GE.NMX) GO TO 838
107*     GO TO 837
108*     838 TAB=L-1
109*     NMX=NUMLY
110*     837 CONTINUE
111*     C
112*     C IDENTIFY MEMBERS OF THE TERMINAL APPENDAGE SET
113*     C
114*     DO 898 L=1,NH
115*     898 TIL=0
116*     IF(TAB.EQ.NH) GO TO 840
117*     DO 839 J=1,NH
118*     K=TAMF(J)
119*     IF(H(K).EQ.TAB) GO TO 829
120*     GO TO 839
121*     829 CONTINUE
122*     DO 888 L=2,NB
123*     IF(EPSIK,L).EQ.1) TIL=1
124*     888 CONTINUE
125*     839 CONTINUE
126*     840 CONTINUE
127*     C
128*     C INITIAL CALCULATION OF BARYCENTER VECTORS W.R.T. BODY C.G.S
129*     C AND HINGE POINTS
130*     C
131*     IXX(1)=MB(1)
132*     IYY(1)=MB(2)
133*     IZZ(1)=MB(3)
134*     IXV(1)=MB(4)
135*     IX2(1)=MB(5)

```

```

136*      IY2(I)=MB(6)
137*      BMASS(I)=MB(7)
138*      TM=BMASS(1)
139*      DO 35 J=2,NB
140*      IXX(IJ)=MA(IJ-1,1)
141*      IYY(IJ)=MA(IJ-1,2)
142*      IZZ(IJ)=MA(IJ-1,3)
143*      IXV(IJ)=MA(IJ-1,4)
144*      IXZ(IJ)=MA(IJ-1,5)
145*      IYZ(IJ)=MA(IJ-1,6)
146*      BMASS(IJ)=MA(IJ-1,7)
147*      35   IM=TM+BMASS(J)
148*      DO 149 I=1,NB
149*      I1=I-1
150*      DO 149 J=1,NB
151*      J1=J-1
152*      IF(I,EQ,J) GO TO 163
153*      IF(I,GT,J) GO TO 70
154*      IF(I,EO,J) GO TO 80
155*      IF(IEPS(I1,J).EQ.1) GO TO 400
156*      70   LX(I,J)=PA(I1,I1,1)
157*      LY(I,J)=PA(I1,I1,2)
158*      LZ(I,J)=PA(I1,I1,3)
159*      GO TO 149
160*      400  CONTINUE
161*      DO 600 K=I,J
162*      IF(IEPS(IK,J).EQ.1) GO TO 500
163*      600  CONTINUE
164*      GO TO 149
165*      500  LX(I,J)=PA(I1,K,1)
166*      LY(I,J)=PA(I1,K,2)
167*      LZ(I,J)=PA(I1,K,3)
168*      GO TO 149
169*      80   DO 90 L=1,J1
170*      IF(IEPS(IL,J).EQ.1) GO TO 1D
171*      90   CONTINUE
172*      GO TO 149
173*      101  LX(I,J)=PBIL,1)
174*      LY(I,J)=PBIL,2)
175*      LZ(I,J)=PBIL,3)
176*      GO TO 149
177*      163  LX(I,J)=0.
178*      LY(I,J)=0.
179*      LZ(I,J)=0.
180*      149  CONTINUE
181*      DO 13 N=1,NB
182*      DO 13 J=1,NB
183*      DX(N,J)=LXIN+J1
184*      DY(N,J)=LYIN+J1
185*      DZ(N,J)=LZIN+J1
186*      DO 13 K=1,NB
187*      DXIN+J)=DX(N,J)-(BMASS(K)/TM)*LXIN+K)
188*      DYIN+J)=DY(N,J)-(BMASS(K)/TM)*LYIN+K)
189*      13   DZIN+J)=DZ(N,J)-(BMASS(K)/TM)*LZIN+K)
190*      C
191*      C      CALCULATION OF AUGMENTED INERTIA DYADICS FOR EACH BODY
192*      C
193*      DO 31 N=1,NB
194*      PH(N+1,1)=IXX(N)
195*      PH(N+1,2)=-IYX(N)
196*      PH(N+1,3)=-IZX(N)
197*      PH(N+2,1)=IYX(N)
198*      PH(N+2,2)=-IZX(N)
199*      PH(N+3,1)=IZZ(N)
200*      DO 30 J=1,NB
201*      PH(N+1,1)=PH(N+1,1)+BMASS(J)*(DX(N,J)**2+DZ(N,J)**2)
202*      PH(N+1,2)=PH(N+1,2)-BMASS(J)*DX(N,J)*DY(N,J)
203*      PH(N+1,3)=PH(N+1,3)-BMASS(J)*DX(N,J)*DZ(N,J)
204*      PH(N+2,1)=PH(N+2,2)+BMASS(J)*(DX(N,J)**2+DZ(N,J)**2)
205*      PH(N+2,2)=PH(N+2,2)-BMASS(J)*DX(N,J)*DZ(N,J)
206*      30   PH(N+3,1)=PH(N+3,1)+BMASS(J)*(DX(N,J)**2+DY(N,J)**2)

```

```

207*      PH(N,2+1)=PH(N,1+2)
208*      PH(N,3+1)=PH(N,1+3)
209*      31      PH(N,3+2)=PH(N,2+3)
210*      DO 571 R=1,NB
211*      DO 571 J=1,3
212*      DO 571 K=1,3
213*      571      PS(R,R,J,K)=PH(R,J,K)
214*      DO 763 J=1,NH
215*      G15(J)=G1J,1)**2
216*      G25(J)=G1J,2)**2
217*      G35(J)=G1J,3)**2
218*      G12(J)=G1J,1)*G1J,2
219*      G13(J)=G1J,1)*G1J,3
220*      763      G23(J)=G1J,21*G1J,31
221*      DO 3700 R=2,NB
222*      DO 3700 S=2,NB
223*      IF(R.EQ.S) GO TO 3700
224*      IF(III(R-1).NE.1).OR.(II(S-1).NE.1)) GO TO 3700
225*      DX2=DX(R,S)+DXIS,R1*TM
226*      DY2=DY(R,S)+DYIS,R1*TM
227*      DZ2=DZ(R,S)+DZIS,R1*TM
228*      PS(R,S+1,1)=-DY2-DZ2
229*      PS(R,S+1,2)=DXIS,R1+DYIR,S1*TM
230*      PS(R,S+1,3)=DXIS,R1+DZIR,S1*TM
231*      PS(R,S+2,1)=DYIS,R1+DXIR,S1*TM
232*      PS(R,S+2,2)=DX2-DZ2
233*      PS(R,S+2,3)=DYIS,R1+DZIR,S1*TM
234*      PS(R,S+3,1)=DZIS,R1+DXIR,S1*TM
235*      PS(R,S+3,2)=DZIS,R1+DYIR,S1*TM
236*      PS(R,S+3,3)=-DX2-DY2
237*      3700      CONTINUE
238*      IT=0
239*      DO 1401 I=1,NH
240*      IF(II(I).NE.1) GO TO 1401
241*      II=II+1
242*      IF(PI(I).EQ.1) IT=II-1
243*      KT=0
244*      DO 1400 K=1,NH
245*      IF(II(K).NE.1) GO TO 1400
246*      KT=KT+1
247*      IF(PI(K).EQ.1) KT=KT-1
248*      IF(I,G1,K) GO TO 1400
249*      DO 1410 J=1,3
250*      1410      CW(J)=0,
251*      DO 1800 R=2,NB
252*      DO 1800 S=2,NB
253*      IF((EPS1K,S1).EQ.0).OR.(EPS1I,R1).EQ.0)) GO TO 1800
254*      DO 1802 J=1,3
255*      CE(J)=0.
256*      DO 1802 L=1,3
257*      1802      CE(J)=CE(J)+PS(R,S+J,L)*G1K,L)
258*      DO 1430 J=1,3
259*      1430      CW(J)=CW(J)+CE(J)
260*      1800      CONTINUE
261*      ASB(I,K)= G(I,1)+CW(1)+G(I,2)+CW(2)+G(I,3)+CW(3)
262*      IF((PI(I),EQ.1).OR.(PI(K),EQ.1)) GO TO 1400
263*      AT(IT,KT)=ASB(I,K)
264*      AT(K1,IT)=AT(IT,KT)
265*      1400      CONTINUE
266*      1401      CONTINUE
267*      NMT=KT
268*      C
269*      C      INVERT THE TERMINAL APPENDAGE MATRIX OF CONSTANTS
270*      C
271*      IF(NMT.EQ.0) GO TO 1422
272*      CALL AINVDT(AT,Q,NMT,$1422+WRK)
273*      1422      CONTINUE
274*      RETURN
275*      ENTRY MRATE(NH,1H,TB,TA,FB,FA,GM,GMD,GHDD,W0,EC)
276*      DIMENSION TB(3),TA(NH,3),FB(3),FA(NH,3),GM(11,6MD11),GMD(11),
277*      $W0(31*WX01U),WY0(U1),WZ0(U1),E1S3,11,TH(NH)

```

```

278*      DOUBLE PRECISION ECV)
279*      C      BODY-TO-BODY COORDINATE TRANSFORMATION MATRICES
280*      C
281*      C      DO 335 J=1,NH
282*      C      K=H(J)
283*      C      IF(LI(J).EQ.1) GO TO 5021
284*      C      SGM=SIN(GM(J))
285*      C      CGM=SQRT(1.-SGM**2)
286*      C      IF(ABS(GM(J)).LT.(1.570796327)) GO TO 5020
287*      C      IF(ABS(GM(J)),GT,(4.712388981)) GO TO 5019
288*      C      CGM=-CGM
289*      C      GO TO 5020
290*      5019  I=(ABS(GM(J)),GT,(7.85398)) CGM=COS(GM(J))
291*      5020  CONTINUE
292*      C      GM1=1.-CGM
293*      C      G12CG=G12(J)*CGM1
294*      C      G13CG=G13(J)*CGM1
295*      C      G23CG=G23(J)*CGM1
296*      C      G1SG=G1(J+1)*SGM
297*      C      G2SG=G2(J+2)*SGM
298*      C      G3SG=G3(J+3)*SGM
299*      C      BA(1,1)=CGM*G1S(J)*CGM1
300*      C      BA(1,2)=G3SG*G12CG
301*      C      BA(1,3)=-G2SG*G13CG
302*      C      BA(2,1)=-G3SG*G12CG
303*      C      BA(2,2)=CGM*G2S(J)*CGM1
304*      C      BA(2,3)=G1SG*G23CG
305*      C      BA(3,1)=G2SG*G13CG
306*      C      BA(3,2)=-G1SG*G23CG
307*      C      BA(3,3)=CGM*G3S(J)*CGM1
308*      C      GO TO 5022
309*      C
310*      5021  CONTINUE
311*      C      BA(1,1)=1.
312*      C      BA(1,2)=G1(J+3)*GM(J)
313*      C      BA(1,3)=-G1(J+2)*GM(J)
314*      C      BA(2,1)=-BA(1,2)
315*      C      BA(2,2)=1.
316*      C      BA(2,3)=G1(J+1)*GM(J)
317*      C      BA(3,1)=-BA(1,3)
318*      C      BA(3,2)=-BA(2,3)
319*      C      BA(3,3)=1.
320*      C
321*      5022  CONTINUE
322*      C      IF(I1.EQ.0) GO TO 3350
323*      C      DO 334 L=1,3
324*      C      DO 334 M=1,3
325*      C      T(1,J+1,L,M)=0.
326*      334   T(1,J+1,L,M)=T(1,J+1,L,M)+BA(I,L)*T(I,K+1,I,M)
327*      C      GO TO 335
328*      C
329*      3350  CONTINUE
330*      C      DO 3351 L=1,3
331*      C      DO 3351 M=1,3
332*      3351  T(1,J+1,L,M)=BA(L,M)
333*      C
334*      C      COORD. TRANSFORMATION OF G VECTORS INTO REF. BODY FRAME
335*      C
336*      C      DO 362 I=1,NH
337*      C      DO 362 J=1,3
338*      C      GO(I,J)=0.
339*      C      DO 362 K=1,3
340*      C      GO(I,J)=GO(I,J)+T(I,I+K,J)*G(I,K)
341*      362   CONTINUE
342*      C
343*      C      ANG. VELOCITY COMPONENTS OF EACH BODY (IN REF. BODY FRAME)
344*      C
345*      C      DO 361 J=1,NB
346*      C      WX0(J)=W0(1)
347*      C      WY0(J)=W0(2)
348*      C      WZ0(J)=W0(3)
349*      C      DO 36 K=1,NH

```

```

350*      IF(EPS(IK+J),EQ.,0) GO TO 36
351*      WX0(J)=WX0(IJ)+GMD(K)*GO(IK+1)
352*      WY0(J)=WY0(IJ)+GMD(K)*GO(IK+2)
353*      WZ0(J)=WZ0(IJ)+GMD(K)*GO(IK+3)
354*      36      CONTINUE
355*      361     CONTINUE
356*      C
357*      C      TRANSFORM AUGMENTED BODY INERTIA DYADICS TO REF. BODY FRAME
358*      C
359*      DO 363 I=2,NB
360*      DO 364 J=1,3
361*      DO 364 K=1,3
362*      AB(I,J,K)=0.
363*      DO 364 L=1,3
364*      AB(I,J,K)=AB(I,J,K)+PH(I,J,L)*T(I,I,L+K)
365*      364     CONTINUE
366*      DO 365 J=1,3
367*      DO 365 K=1,3
368*      PS(I,I,J,K)=0.
369*      DO 365 L=1,3
370*      PS(I,I,J,K)=PS(I,I,J,K)+T(I,I,L,J)*AB(I,L,K)
371*      365     CONTINUE
372*      363     CONTINUE
373*      DO 471 J=1,3
374*      DO 471 K=1,3
375*      471     PS(I,I,J,K)=PH(I,J,K)
376*      FEZO(I)=FB(I1)
377*      FEYO(I)=FB(I2)
378*      FEZO(I)=FB(I3)
379*      TX0(I)=TB(I1)
380*      TY0(I)=TB(I2)
381*      TZ0(I)=TB(I3)
382*      DO 17 I=2,NB
383*      K=I-1
384*      FEZO(I)=T(I,I,I+1)+FA(K,1)*T(I,I+2,1)+FA(K,2)*T(I,I+3,1)+FA(K,3)
385*      FEYO(I)=T(I,I,I+2)+FA(K,1)*T(I,I+2,2)+FA(K,2)*T(I,I+3,2)+FA(K,3)
386*      FEZO(I)=T(I,I,I+3)+FA(K,1)*T(I,I+2,3)+FA(K,2)*T(I,I+3,3)+FA(K,3)
387*      TX0(I)=T(I,I,I+1)+TA(K,1)*T(I,I+2,1)+TA(K,2)*T(I,I+3,1)+TA(K,3)
388*      TY0(I)=T(I,I,I+2)+TA(K,1)*T(I,I+2,2)+TA(K,2)*T(I,I+3,2)+TA(K,3)
389*      TZ0(I)=T(I,I,I+3)+TA(K,1)*T(I,I+2,3)+TA(K,2)*T(I,I+3,3)+TA(K,3)
390*      DO 17 J=1,NB
391*      DX0(I,J)=T(I,I,I+1)*DX(I,J)+T(I,I+2,1)*DY(I,J)+T(I,I+3,1)*DZ(I,J)
392*      DY0(I,J)=T(I,I,I+2)*DX(I,J)+T(I,I+2,2)*DY(I,J)+T(I,I+3,2)*DZ(I,J)
393*      DZ0(I,J)=T(I,I,I+3)*DX(I,J)+T(I,I+2,3)*DY(I,J)+T(I,I+3,3)*DZ(I,J)
394*      17      CONTINUE
395*      DO 367 I=1,NB
396*      DX0(1,I)=DX(1,I)
397*      DY0(1,I)=DY(1,I)
398*      367     DZ0(1,I)=DZ(1,I)
399*      C
400*      C      ADDITIONAL AUGMENTED INERTIA DYADICS (IN REF.BODY FRAME)
401*      C
402*      DO 37 I=1,NB
403*      DO 37 J=1,NB
404*      IF(I,J,EQ.,J) GO TO 37
405*      PS(I,J,I+1)=-TM*(DX0(I,J)*DY0(I,J)+DZ0(I,J,I))+DZ0(I,J,I)
406*      PS(I,J,I+2)=TM*(DX0(I,J)*DY0(I,J)+DZ0(I,J,I))
407*      PS(I,J,I+3)=TM*(DX0(I,J)*DZ0(I,J,I))
408*      PS(I,J,I+2)=TM*(DY0(I,J)*DX0(I,J))
409*      PS(I,J,I+2)=TM*(DX0(I,J)*DX0(I,J)+DZ0(I,J,I))+DZ0(I,J,I)
410*      PS(I,J,I+3)=TM*(DY0(I,J)*DZ0(I,J,I))
411*      PS(I,J,I+3)=TM*(DZ0(I,J,I)*DX0(I,J))
412*      PS(I,J,I+2)=TM*(DZ0(I,J,I)*DY0(I,J))
413*      PS(I,J,I+3)=TM*(DX0(I,J,I)*DY0(I,J)+DY0(I,J,I)+DY0(I,J,I))
414*      37      CONTINUE
415*      C
416*      C      VECTOR CROSS PRODUCTS DESCRIBING SYSTEM ROTATIONAL COUPLING.
417*      C      (QUADRATIC TERMS INVOLVING THE CONNECTING BODY ANGULAR
418*      C      VELOCITIES AND THE MUTUAL BARYCENTER-HINGE VECTORS)
419*      C
420*      DO 230 N=1,NB

```

```

421*      CPX=0.
422*      CPY=0.
423*      CPZ=0.
424*      DO 2301 L=1,NB
425*      IF(N.EQ.L) GO TO 2301
426*      WDX=WYO(L)+DZO(L,N)-WZ0(L)+DY0(L,N)
427*      WDY=WZO(L)+DX0(L,N)-WX0(L)+DZO(L,N)
428*      WDZ=WX0(L)+DY0(L,N)-HY0(L)+DX0(L,N)
429*      WWDX=TM*(WYO(L)+WZ0(L)+WDY)+FEX0(L)
430*      WWDY=TM*(WZO(L)+WDX-WX0(L)+WDZ)+FEY0(L)
431*      WWDZ=TM*(WX0(L)+WDY-WY0(L)+WDX)+FEZ0(L)
432*      DWWDX=DY0(N,L)+WWDZ-DZO(N,L)+WWDY
433*      DWWDY=DZO(N,L)+WWDX-DX0(N,L)+WWDZ
434*      DWWDZ=DX0(N,L)+WWDY-DY0(N,L)+WWDX
435*      CPX=CPX+DWWDX
436*      CPY=CPY+DWWDY
437*      CPZ=CPZ+DWWDZ
438*      2301 CONTINUE
439*      DX=DY0(N,N)+FEZ0(N)-DZO(N,N)+FEY0(N)
440*      DY=DZO(N,N)+FEX0(N)-DX0(N,N)+FEZ0(N)
441*      DZ=DX0(N,N)+FEY0(N)-DY0(N,N)+FEX0(N)
442*      HX=PS(N,N+1+1)*WX0(N)+PS(N,N+1+2)*WY0(N)+PS(N,N+1+3)*WZ0(N)
443*      HY=PS(N,N+2+1)*WX0(N)+PS(N,N+2+2)*WY0(N)+PS(N,N+2+3)*WZ0(N)
444*      HZ=PS(N,N+3+1)*WX0(N)+PS(N,N+3+2)*WY0(N)+PS(N,N+3+3)*WZ0(N)
445*      K = 3*N-1
446*      EIK+1,1)=HY*WZ0(N)-HZ*WY0(N)+TX0(N)+CPX+DFX
447*      EIK+2,1)=HZ*WX0(N)-HX*WZ0(N)+TY0(N)+CPY+DFY
448*      EIK+3,1)=HX*WY0(N)-HY*WX0(N)+TZ0(N)+CPZ+DFZ
449*      230 CONTINUE
450*      C
451*      C      ADD MATRIX ELEMENT COMPUTATION (3X3)
452*      C
453*      DO 3001 I=1,3
454*      DO 3001 J=1,3
455*      3001 ADD(I,J)=0.
456*      DO 3 I=1,NB
457*      DO 3 J=1,NB
458*      ADD(1,1)=ADD(1,1)+PS(I,J+1+1)
459*      ADD(1,2)=ADD(1,2)+PS(I,J+1+2)
460*      ADD(1,3)=ADD(1,3)+PS(I,J+1+3)
461*      ADD(2,1)=ADD(2,1)+PS(I,J+2+1)
462*      ADD(2,2)=ADD(2,2)+PS(I,J+2+2)
463*      ADD(2,3)=ADD(2,3)+PS(I,J+2+3)
464*      ADD(3,1)=ADD(3,1)+PS(I,J+3+1)
465*      3 CONTINUE
466*      ADD(2,1)=ADD(1,2)
467*      ADD(3,1)=ADD(1,3)
468*      ADD(3,2)=ADD(2,3)
469*      C
470*      C      ADD VECTOR ELEMENT COMPUTATION (3X1)
471*      C
472*      DO 7 K=1,NH
473*      AVIK,1=0.
474*      AVIK,2=0.
475*      AVIK,3=0.
476*      DO 7 I=1,N3
477*      DO 7 J=1,NB
478*      IF(EPS(K,J).EQ.0) GO TO 7
479*      DO 10 M=1,3
480*      10 AVIK,M)=AVIK,M1+PS(I,J,M+L)+GO(K,L)
481*      7 CONTINUE
482*      C
483*      C      AKM SCALAR ELEMENT COMPUTATION
484*      C
485*      DO 14 K=1,NH
486*      DO 14 M=1,NH
487*      IF(K.GT.M) GO TO 14
488*      IF((I1(K).EQ.1).AND.(I1(M).EQ.1)) GO TO 1399
489*      AIS11=0.
490*      AIS12=0.
491*      AIS13=0.

```

```

492*      DO 15 J=1,NB
493*      DO 15 I=1,NB
494*      IF((EPS(K,J),EQ,01),OR,(EPS(M,I),EQ,01)) GO TO 15
495*      DO 18 N=1,3
496*      DO 18 L=1,3
497*      18 AIS(N)=AIS(N)*PS(I,J,N,L)*GO(M,L)
498*      15 CONTINUE
499*      AIS(K,M)=GO(K,1)*AIS(1)+GO(K,2)*AIS(2)+GO(K,3)*AIS(3)
500*      GO TO 14
501*      1399 AIS(K,M)=AS3(K,M)
502*      14 CONTINUE
503*      C
504*      C COMPUTE CORRECTION ELEMENTS FOR IEI VECTOR
505*      C
506*      DO 40 I=1,NB
507*      EA(1)=0.
508*      EA(2)=0.
509*      EA(3)=0.
510*      DO 41 J=1,NB
511*      DO 411 M=1,3
512*      411 CW(M)=0.
513*      DO 42 K=1,NH
514*      IF(EPS(K,J),EQ,0) GO TO 42
515*      K9=KK(K)
516*      CW(1)=CW(1)+GMD(K)*IWY0(KB)*GO(K,3)-WZ0(KB)*GO(K,2)
517*      CW(2)=CW(2)+GMD(K)*IWZ0(KB)*GO(K,1)-WX0(KB)*GO(K,3)
518*      CW(3)=CW(3)+GMD(K)*INX0(KB)*GO(K,2)-WY0(KB)*GO(K,1)
519*      42 CONTINUE
520*      DO 45 M=1,3
521*      DO 45 L=1,3
522*      45 EA(M)=EA(M)*PS(I,J,M,L)*CW(L)
523*      41 CONTINUE
524*      K1=3*(I-1)
525*      E(K1+1)=E(K1+1)-EA(1)
526*      E(K1+2)=E(K1+2)-EA(2)
527*      E(K1+3)=E(K1+3)-EA(3)
528*      40 CONTINUE
529*      DO 55 M=1,3
530*      55 EC(IM)=E(M,I)
531*      DO 52 J=2,NB
532*      DO 52 M=1,3
533*      K1=3*(J-1)+M
534*      52 EC(M)=EC(M)+E(K1)
535*      I=0
536*      DO 60 K=1,NH
537*      IF(PI(K),NE,0) GO TO 60
538*      I=I+1
539*      EC(I)=0.
540*      DO 601 M=1,3
541*      601 CS(M)=0.
542*      DO 61 J=1,NB
543*      IF(EPS(K,J),EQ,0) GO TO 61
544*      DO 65 M=1,3
545*      J1=3*(J-1)+M
546*      65 CE(M)=CE(M)+E(J1)
547*      61 CONTINUE
548*      DO 66 L=1,3
549*      66 EC(I)=EC(I)+GO(K,L)*CE(L)
550*      EC(I)=EC(I)*TH(K)
551*      60 CONTINUE
552*      DO 610 I=1,3
553*      DO 610 J=1,NH
554*      IF(PI(J),EQ,0) GO TO 610
555*      EC(I)=EC(I)-AV(J,I)*GMDD(J)
556*      610 CONTINUE
557*      K=0
558*      IV=3
559*      DO 612 I=1,NH
560*      IF(PI(I),NE,0) GO TO 612
561*      K=K+1
562*      IV=IV+1
563*      DO 511 J=1,NH

```

```

564*      IF(PI(I,J).EQ.0) GO TO 611
565*      IF(I.GT.J) AS(I,J)=AS(J,I)
566*      ECIK+3)=ECIK+3)-AS(I,J)*GMDD(I,J)
567*      611 CONTINUE
568*      612 CONTINUE
569*      C
570*      C      LOAD SYSTEM MATRIX (A) WITH ADD+ADK+AKM ELEMENTS
571*      C
572*      DO 23 I=1,3
573*      DO 23 J=1,3
574*      23 A(I,J)=ADD(I,J)
575*      DO 24 I=1,3
576*      K=0
577*      DO 24 J=1,NH
578*      IF(PI(I,J).NE.0) GO TO 24
579*      K=K+1
580*      A(K+3,I)=AV(J,I)
581*      A(I,K+3)=AV(J,I)
582*      24 CONTINUE
583*      K=0
584*      DO 250 I=1,NH
585*      IF(PI(I,I).NE.0) GO TO 250
586*      K=K+1
587*      L=0
588*      DO 25 J=1,NH
589*      IF(PI(J,J).NE.0) GO TO 25
590*      L=L+1
591*      IF(L.GT.L) GO TO 26
592*      A(K+3,L+3)=AS(I,J)
593*      GO TO 25
594*      26 A(K+3,L+3)=A(L+3,K+3)
595*      25 CONTINUE
596*      250 CONTINUE
597*      C
598*      C      SOLVE SYSTEM MATRIX FOR REFERENCE BODY ANG. ACCELERATION AND HINGE
599*      C      (RELATIVE) ROTATIONAL ACCELERATIONS
600*      C
601*      C      EXCHANGE ROWS IN A-MATRIX AND RT.HAND SIDE VECTOR
602*      C
603*      DO 619 K=1,3
604*      EEIKI)=ECIKI
605*      DO 619 I=1,IV
606*      619 AEIK+I)=AIK,I)
607*      L=0
608*      DO 620 K=1,NH
609*      IF(TII(K).EQ.1) GO TO 620
610*      L=L+1
611*      EEIL+3)=ECIK+3)
612*      DO 621 I=1,IV
613*      621 AEIL+3,I)=AIK+3,I)
614*      620 CONTINUE
615*      DO 630 K=1,NH
616*      IF(TII(K).EQ.0) GO TO 630
617*      L=L+1
618*      EEIL+3)=ECIK+3)
619*      DO 631 I=1,IV
620*      631 AEIL+3,I)=AIK+3,I)
621*      630 CONTINUE
622*      C
623*      C      EXCHANGE COLUMNS IN A-MATRIX
624*      C
625*      DO 639 K=1,3
626*      DO 639 I=1,IV
627*      639 A(I,K)=AEII,K)
628*      L=0
629*      DO 640 K=1,NH
630*      IF(TII(K).EQ.1) GO TO 640
631*      L=L+1
632*      DO 641 I=1,IV
633*      641 AI,I,L+3)=AEII,K+3)
634*      640 CONTINUE
635*      DO 650 K=1,NH

```

```

636*      IF(TII(K).EQ.0) GO TO 650
637*      L=L+1
638*      DO 651 I=1,IV
639*      651 A(I,L+3)=AE(I,K+3)
640*      650 CONTINUE
641*      C
642*      C      SOLVE FOR HINGE-ANGLE ACCELERATIONS (NON-TERM. APPENDAGE)
643*      C
644*      NL=IV-NMT
645*      IF(NMT.EQ.0) GO TO 6561
646*      DO 652 I=1,NL
647*      DO 652 K=1,NMT
648*      AL(I,K)=0.
649*      DO 652 L=1,NMT
650*      LL=NL+L
651*      652 AL(I,K)=AL(I,K)+A(I,LL)*AT(L,K)
652*      DO 654 I=1,NL
653*      DO 654 K=1,NL
654*      AE(I,K)=0.
655*      DO 653 L=1,NMT
656*      LL=NL+L
657*      653 AE(I,K)=AE(I,K)+AL(I,L)*A(LL,K)
658*      654 A(I,K)=A(I,K)-AE(I,K)
659*      DO 656 I=1,NL
660*      EC(I)=0.
661*      DO 655 L=1,NMT
662*      LL=NL+L
663*      EC(I)=EC(I)+AL(I,L)*EE(LL)
664*      EE(I)=EE(I)-EC(I)
665*      6561 CONTINUE
666*      CALL SODIA(V,NL,EE,V,I,$92,WPK)
667*      C
668*      C      SOLVE FOR HINGE-ANGLE ACCELERATIONS (TERM. APPENDAGE)
669*      C
670*      IF(NMT.EQ.0) GO TO 6581
671*      DO 657 I=1,NMT
672*      II=NL+I
673*      EC(I)=EE(II)
674*      DO 657 L=1,NL
675*      657 EC(I)=EC(II)-A(II,L)*EE(L)
676*      DO 658 I=1,NMT
677*      II=NL+I
678*      EE(II)=0.
679*      DO 658 L=1,NMT
680*      658 EE(II)=EE(II)+AT(I,L)*EC(L)
681*      6581 CONTINUE
682*      C
683*      C      RECONSTRUCT SOLUTION VECTOR IN ORIGINAL ORDER
684*      C
685*      EC(1)=EE(1)
686*      EC(2)=EE(2)
687*      EC(3)=EE(3)
688*      L=3
689*      DO 659 K=1,NH
690*      IF((TII(K).EQ.1).OR.(PIIK).EQ.1) GO TO 659
691*      L=L+1
692*      EC(K+3)=EE(L)
693*      659 CONTINUE
694*      L=NL
695*      DO 661 K=1,NH
696*      IF((TII(K).NE.1).OR.(PIIK).EQ.1) GO TO 661
697*      L=L+1
698*      EC(K+3)=EE(L)
699*      661 CONTINUE
700*      DO 662 K=1,NH
701*      IF(PIIK).NE.1 GO TO 662
702*      EC(K+3)=GMDO(K)
703*      662 CONTINUE
704*      92 CONTINUE
705*      RETURN
706*      END

```

References

1. Hooker, W. W., and Margulies, G., "The Dynamical Attitude Equations for an n-Body Satellite," *J. Astronaut. Sci.*, Vol. 12, pp. 123-128, 1965.
2. Russell, W. J., *On the Formulation of Equations of Rotational Motion for an N-Body Spacecraft*, TR-0200 (4133)-2, Aerospace Corp., El Segundo, Calif., Feb. 1969.
3. Velman, J. R., "Simulation Results for a Dual-Spin Spacecraft," *Proceedings of the Symposium on Attitude Stabilization and Control of Dual-Spin Spacecraft*, Rept. SAMSO-TR-68-191, Nov. 1967.
4. Fleischer, G. E., *Multi-Rigid-Body Attitude Dynamics Simulation*, Technical Report 32-1516, Jet Propulsion Laboratory, Pasadena, Calif., Feb. 15, 1971.
5. Farrell, J. L., and Newton, J. K., "Continuous and Discrete RAE Structural Models," *J. Spacecraft and Rockets*, Vol. 6, pp. 414-423, 1969.
6. Fleischer, G. E., and McGlinchey, L. F., *Viking Thrust Vector Control Dynamics Using Hybrid Coordinates to Model Vehicle Flexibility and Propellant Slosh*, AAS Paper No. 71-348, presented at AAS/AIAA Astrodynamics Specialists Conference 1971, Ft. Lauderdale, Florida, Aug. 17-19, 1971.
7. Fleischer, G. E., "Simulation of Flexible Appendage Interactions with Mariner Venus/Mercury Attitude Control and Science Platform Pointing," presented at AAS/AIAA Astrodynamics Conference, Vail, Colo., July 16-18, 1973.
8. Hooker, W. W., "A Set of Dynamical Attitude Equations for an Arbitrary n-Body Satellite Having r Rotational Degrees of Freedom," *AIAA J.*, Vol. 8, pp. 1205-1207, 1970.
9. Likins, P. W., and Fleischer, G. E., *Large Deformation Modal Coordinates for Nonrigid Vehicle Dynamics*, Technical Report 32-1565, Jet Propulsion Laboratory, Pasadena, Calif., Nov. 1, 1972.